PROYECTO INTEGRADOR CARRERA DE INGENIERÍA MECÁNICA

DETECCIÓN TEMPRANA DE DESVIACIONES DEL COMPORTAMIENTO NOMINAL DE SISTEMAS UTILIZANDO ALGORITMOS DE MACHINE LEARNING

Uriel A. Muñoz
Ingeniería Mecánica

Ing. José Relloso
Director

Dr. Félix Rojo
Co-director

Miembros del Jurado
Dr. Jorge Osmar Lugo (Instituto Balseiro / INVAP)
Dr. Eugenio Urdapilleta (Instituto Balseiro)

Junio de 2019

INVAP S.A.

Instituto Balseiro
Universidad Nacional de Cuyo
Comisión Nacional de Energía Atómica
Argentina

A mis padres, Isabel y Mauricio
A mi familia
A Sara

Índice de contenidos

Índice de figuras

Índice de tablas

Resumen

En sistemas tan complejos como los del campo aeroespacial, los sub-sistemas se diseñan para minimizar la inferencia mutua. Sin embargo, anomalías transversales a varios sub-sistemas existen, y son difíciles de detectar y entender. Sistemas de detección de anomalías integrales juegan un papel crítico en estas situaciones. En este trabajo se presentan herramientas de *Machine Learning* para la detección temprana de anomalías en una plataforma del área aeroespacial. Los métodos utilizados son: *Gaussian Mixture Model*, *Principal Component Classifier* y *Forecasting*. Este último tiene el propósito de análisis de variables individuales, mientras que los otros dos tienen un espectro de aplicación integral, donde se apunta a la detección de cambios en la estructura de los datos y en menor medida a valores extremos individuales. En todos los casos son herramientas que se pensaron para ayudar a complementar el análisis del profesional (experto de dominio) en su trabajo, y no ser utilizadas independientemente. Son herramientas que proveen versatilidad en el análisis y permiten que se puedan aplicar ágilmente a distintos conjuntos de datos. Se lograron detectar anomalías artificiales de forma satisfactoria, para casos puntuales e integrales.

Palabras clave: MACHINE LEARNING, ANOMALÍAS, ANÁLISIS COMPONENTES PRINCIPALES, CLUSTERING, FORECASTING

Abstract

In complex system like those from the aerospace field, subsystems (atomic constituents of the full system) are designed to minimize or mitigate mutual inference. However, anomalies usually emerge as a collective phenomenon which turns the detection and the isolation a difficult task. Comprehensive anomalies detection systems play a critical role in these situations. In this thesis we present *Machine Learning* tools for the early detection of anomalies in a platform of the aerospace area. The methods used are: *Gaussian Mixture Model*, *Principal Component Classifier* and *Forecasting*. The latter has the purpose of analyzing individual variables, while the first two have a integral approach, where the objetive is to detect structure changes of the data and not so much extreme values. In all cases they are tools to help complement the analysis of the expert professional in their work, and not to be used autonomously. They are tools that demonstrate versatility in the analysis and allow to be applied agilely to different data sets. It was possible to detect artificial anomalies satisfactorily, for specific and integral cases.

Keywords: MACHINE LEARNING, ANOMALIES, NOVELTY DETECTION, PRINCIPAL COMPONENT ANALYSIS, CLUSTERING, FORECASTING

Capítulo 1

Introducción

"La primera tarea de la educación es agitar la vida, pero dejarla libre para que se desarrolle"
— María Montessori, 1870-1952

1.1. Motivación

Gran cantidad de la tecnología utilizada cotidianamente genera un cúmulo enorme de datos que son susceptibles de ser analizados con algoritmos de *Machine Learning*[2, 3]. La utilización de estos algoritmos ha tenido un crecimiento marcado en los últimos años en campos muy diversos como el análisis de imágenes de cultivos o análisis bursátiles.

Actualmente existe una fuerte tendencia hacia el campo de la ingeniería de mantenimiento. Los algoritmos de *Machine Learning* se incorporaron como una herramienta extra en el mantenimiento predictivo. Esto permite la caracterización del comportamiento de los aparatos y hace posible realizar predicciones sobre el mismo. Esto permite conocer los límites de funcionamiento del instrumento, conocer señales tempranas de fatiga y, consecuentemente, evitar daños mayores derivados de un diagnóstico incorrecto o fuera de tiempo.

El objetivo de este proyecto integrador es identificar e implementar algoritmos de aprendizaje para la detección temprana de anomalías. Los algoritmos se evalúan utilizando una plataforma proveniente del área aeroespacial provista por la empresa INVAP.

1.2. Detección de anomalías

La detección de anomalías se encarga de reconocer valores, o conjuntos de valores, que difieren de la tendencia nominal de un determinado conjunto de datos. Mientras

que varios sistemas modernos la utilizan en la detección de anomalías, la mayor parte de ellos involucra conocimiento *a priori* del sistema a analizar. El hecho de tener que conocer el sistema en detalle de antemano conlleva la realización de modelos complejos que muchas veces no terminan de representar completamente el sistema. La forma de solucionar esto es *aprender* de los datos generados por el sistema en cuestión, durante el intervalo de tiempo que, por experiencia, el sistema estuvo funcionando nominalmente.

En este proyecto se utilizará una vasta cantidad de datos de telemetría proveniente de un número de sistemas de una plataforma del área aeroespacial. Los datos son una serie temporal y multidimensional producida por los distintos sensores de dicho sistema.

La existencia de anomalías o fallas es prácticamente imposible de evitar, aún incrementando la fiabilidad de los sistemas al máximo posible. En el sector aeroespacial, esto es aún más relevante dado que la distancia a nuestro sistema imposibilita la inspección directa o reparación de la falla. Consecuentemente, una detección temprana de una desviación de los valores nominales es significativamente importante para evitar situaciones catastróficas como lo puede ser la pérdida de control.

En sistemas tan complejos como los del campo aeroespacial, los sub-sistemas se diseñan para minimizar la inferencia mutua. Sin embargo, anomalías transversales a varios sub-sistemas existen, y son difíciles de detectar y entender el fenómeno. Sistemas de detección de anomalías integrales juegan un papel crítico en estas situaciones.

Mientras que la comunicación de telemetría a estaciones terrestres tienen como principal objetivo el análisis manual por parte de expertos, en el último tiempo se ha encontrado que una variedad de técnicas de *Machine Learning* pueden aplicarse a estos datos.

1.3. Data Mining y Machine Learning

Vivimos en una era digital donde la información en forma de datos crudos son generados a escalas astronómicas. El número de ejemplos es extenso, desde que los celulares modernos registran la cantidad de pasos que el usuario da por día, hasta los millones de datos climáticos que se generan por día. Como se mencionó anteriormente, la industria aeroespacial no es la excepción. Las aeronaves y astronaves generan datos sobre el estado de los distintos subsistemas. Estos datos contienen patrones y relaciones que no son visibles mediante una simple inspección manual.

El objetivo de *Data Mining* es extraer datos implícitos, previamente desconocidos, que sean potencialmente útiles. Es un proceso iterativo de creación de un modelo predictivo y descriptivo.

Machine Learning involucra algoritmos que *aprenden* de forma iterativa, de allí una de sus traducciones como *Aprendizaje Automático*, basado en datos previos. Se trata de aprender de datos pasados el fenómeno de trasfondo para poder predecir nuevos

resultados. El conjunto de datos se divide en dos, un conjunto de entrenamiento y un conjunto de testeo. De este último se obtiene una retroalimentación para continuar aprendiendo.

Existen diferentes tipos de algoritmos de aprendizaje que se utilizan para entrenar el modelo. Estos algoritmos se suelen separar en dos grandes categorías: *Aprendizaje Supervisado*, y *Aprendizaje No Supervisado*. Cuál de ellos se utiliza depende del conjunto de datos con los que se trabaja.

1.3.1. Aprendizaje Supervisado

En el paradigma del aprendizaje supervisado se cuenta con datos de aprendizaje que se encuentran etiquetados con "*la respuesta correcta*", los cuales se utilizan para aprender la función que mapea los datos de entrada a los datos de salida. Con esta función de mapeo aprendida, luego se le quieren dar nuevos valores de entrada nunca vistos por la función y predecir nuevos valores de salida.

El termino "*supervisado*"proviene de que, dado un dato de entrenamiento, uno sabe cual debe ser la salida o *output*. Los algoritmos de aprendizaje automatizado se pueden agrupar en algoritmos de *clasificación* y algoritmos de *regresión*.

Los problemas de clasificación son aquellos que tienen definidas distintas clases o categorías a las cuales pertenecen los resultados. Un ejemplo de esto puede ser la clasificación de imágenes de perros según la raza de cada uno. Las clases o categorías serian las distintas razas: *Akita, Boxer, Cocker Spaniel*, entre otras. La forma de entrenamiento es entregarle al modelo una gran cantidad de fotos de perros con sus respectivas etiquetas con el nombre de la raza. Una vez entrenado el sistema, se le puede entregar una foto de un perro, y el sistema devolverá la raza del mismo.

Los problemas de regresión se diferencian a los de clasificación en que la salida o *output* ya no es discreto sino que es continuo y puede obtener valores reales, como puede ser el valor de un voltaje o el valor del dólar. El modelo se entrena y crea una función de ajuste que mapea la entrada a la salida. Luego, puede predecir datos en función de valores nuevos de entrada.

Para la detección de anomalías de forma supervisada se debe contar con ambos casos, el nominal o normal, y casos anómalos. El modelo aprende a partir de los datos de entrenamiento, tratando de encontrar patrones en ambos tipos de comportamiento. El objetivo es encontrar un modelo que haga una clasificación de los nuevos puntos en anómalos o nominales.

1.3.2. Aprendizaje No Supervisado

A diferencia del aprendizaje supervisado, en el paradigma no supervisado los datos de entrada no poseen etiqueta, por lo que no hay datos de salida a los cuales realizar

un mapeo. El objetivo de estos algoritmos es encontrar regularidades o patrones en el espacio de entrada, modelar la estructura y distribución de los datos, y luego evaluar si se puede aprender algo nuevo de los datos.

Este tipo de algoritmos se suele llamar *estimación estadística de densidad*. Uno de los principales métodos es llamado *Clustering*, donde el objetivo es encontrar *"clusters"* o grupos donde se encuentren los valores de entrada. Los *clusters* pueden usarse para clasificación, donde cada grupo que se encuentre define una nueva clase. Se diferencia del caso supervisado en que las nuevas clases se crean mirando datos no etiquetados. Para el caso de detección de anomalías, se comienza una búsqueda de puntos no nominales sin conocimiento previo de los datos. Se supone que las anomalías aparecen suficientemente separadas del resto de los datos. El objetivo es encontrar un modelo que agrupe los datos, bajo alguna métrica adecuada, en distintos *clusters* de instancias nominales. Cuando un punto anómalo aparece, este no pertenece a ningún *cluster* de los encontrados en el entrenamiento.

Estas ideas se representan de forma esquemática en la figura 1.1.

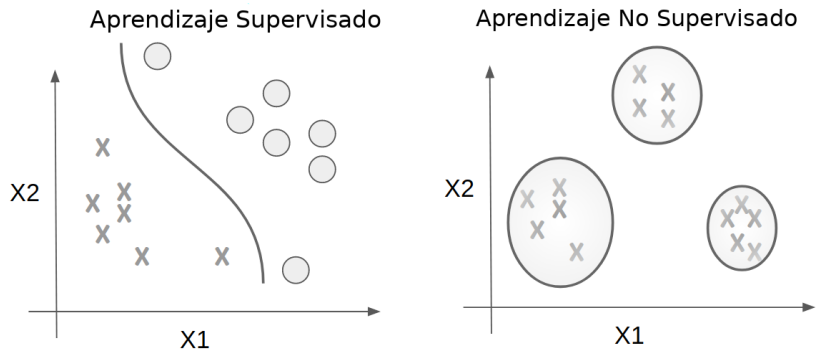


Figura 1.1: Aprendizaje Supervisado versus Aprendizaje No Supervisado.

1.4. Anomalías

Es difícil encontrar algo cuando no se tiene claro qué se está buscando. En general es complejo definir qué es una anomalía, dado que el tipo de anomalía depende de cada problema.

Se puede plantear a una anomalía como un tipo de comportamiento de los datos que difiere, en algún sentido, del comportamiento esperado. El caso más simple es un punto en los datos denominado *outlier*, el cual se diferencia del resto de los datos en base a su valor, u ocurre aleatoria y raramente en comparación con el resto de los puntos. Cuando se habla de anomalía, se abarcan hechos más complejos que solo un valor fuera de lo nominal. Se tienen en cuenta patrones irregulares que muchas veces no pueden ser vistos en una simple inspección.

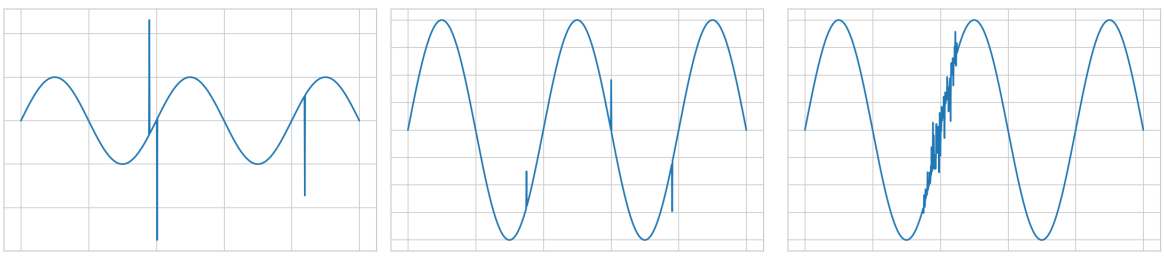
Podemos agrupar a las anomalías en tres clases diferentes según su comportamiento:

* Anomalía Puntual: Usualmente representa algún extremo, irregularidad o desviación que ocurre aleatoriamente o no tiene un significado particular. Es un tipo de anomalía que se conoce como *outlier*. Para entender mejor y lograr una comparación entre los distintos tipos de anomalías, se plantea el siguiente ejemplo: Supongamos que seguimos los movimientos de dinero de la cuenta bancaria de una persona que todos los meses, rigurosamente, paga un alquiler de diez mil pesos y no suele tener gastos puntuales mayores a ese valor. Si detectamos un pago de veinte mil pesos, podríamos considerarlo un *outlier*.

* Anomalía Contextual: Es una instancia que puede ser considerada anómala en un contexto específico. Es decir, que el mismo punto lo consideraríamos anómalo o no según en qué momento o dónde lo encontremos. Para continuar con el ejemplo de la cuenta bancaria, supongamos que esta persona paga el alquiler rigurosamente el décimo día del mes. Sí detectamos un pago de diez mil pesos un día alejado del décimo día podría considerarse una anomalía contextual, dado que, aunque el valor no es anómalo en sí mismo, sí lo es detectarlo un día que no sea el usual.

* Anomalía Colectiva: Se suele representar como un grupo correlacionado, interconectado o secuencial de instancias. Mientras que la ocurrencia de cada punto aislado no es una anomalía en sí misma, la ocurrencia colectiva de ellos sí lo es. Para el caso de la cuenta bancaria podemos considerar una anomalía colectiva un número alto de retiros y depósitos de dinero de valores erráticos que en sí mismos no son altos. Cada valor, al no tener un valor alto, puede pasar como una simple compra; pero que aparezca una rápida sucesión de retiros y depósitos de forma aleatoria en un corto período de tiempo se lo puede considerar como una anomalía colectiva.

Estas ideas se representan de forma esquemática en la figura 1.2.



(a) Anomalía Puntual. **(b)** Anomalía Contextual. **(c)** Anomalía Colectiva.

Figura 1.2: Tipos de anomalías.

1.5. Estructura de Entrenamiento

En esta sección se presentan los pasos que siguen los datos desde su obtención hasta obtener predicciones del modelo. A esta estructura que siguen los datos desde que tienen un formato no estructurado hasta que se obtienen predicciones de ellos, se la llama *pipeline*. El *pipeline* se puede esquematizar según la figura 1.3. En más detalle, se puede describir de la siguiente manera:

1. Obtención de los datos y definición del problema: Este proceso puede darse partiendo desde fuentes muy distintas entre sí. Pueden obtenerse datos a partir de sensores o a partir de encuestas en una red social. En este proyecto integrador este paso se ha realizado previamente en forma parcial. Los datos con los que se trabajan ya están estructurados en una base de datos, por lo que la fase de telecomunicaciones está satisfecha de forma previa.

   La calidad y cantidad de datos que se recopilen será un factor fundamental en el aprendizaje del modelo. Dado que muchos modelos aprenden iterativamente, la cantidad de iteraciones que se pueden hacer dependerá de la cantidad de datos que se tenga. Por otro lado, cuando se habla de la calidad de los datos, se busca representar el espacio de salida o dominio lo más completamente posible. Esto resulta aún más importante en la detección de anomalías dado que si se deja afuera un estado nominal de la plataforma, el modelo solo habrá aprendido parcialmente, y por ende, se habrá perdido confiabilidad en las predicciones.

2. Separación de los datos: Se separan los datos en un conjunto de entrenamiento con los cuales se preparará el modelo, y un conjunto de datos de validación con los cuales se verificará la eficacia del modelo entrenado. La idea principal del conjunto de validación es que estos no hayan participado de ninguna forma en el proceso de entrenamiento. Aunque esto puede parecer directo, es muy simple contaminar los datos desprevenidamente. Un ejemplo simple de contaminación de los datos es normalizar y luego realizar la separación.

   Validar con el conjunto separado nos permite tener una estimación de cómo se comporta el modelo con datos que nunca vio, característica fundamental, dado que el objetivo final es obtener predicciones a partir de nuevos datos. Además, es una herramienta contra el *overfitting*, tema que se verá en el siguiente capítulo.

3. Pre-procesamiento de los datos: Esto involucra todos los procesos necesarios que se le realicen a los datos para poder utilizarlos en el entrenamiento. En este proyecto veremos que algunos de dichos procesos son: encargarnos de qué hacer con los datos nulos o faltantes, descartar variables que no aporten al entrenamiento, normalizar, entre otros.

Un pre-procesamiento que suele resultar muy útil es la reducción de dimensionalidad. La misma suele ser fundamental para afrontar problemas donde las especificaciones de hardware no son suficientes. Algunas variables aportan complejidad al modelo sin mejorar la precisión del mismo, por lo que, a fin de cuentas, se trata de mejorar la relación costo-beneficio. Además, como se verá más adelante, ayuda a tratar la Maldición de la Dimensionalidad.

4. Elección y entrenamiento del modelo: Dependiendo del problema planteado se utilizan unos u otros algoritmos. Por ejemplo, si no se cuenta con datos etiquetados, se enfocará directamente en aprendizaje no supervisado. El objetivo de entrenar el modelo es responder una pregunta o realizar una predicción. Se utilizan los datos para mejorar incrementalmente la habilidad del modelo de realizar el objetivo.

5. Evaluación del modelo: Una vez que se entrenó el modelo, se quiere saber qué tan buenas son las predicciones. En este punto entra en juego el conjunto de validación que se había apartado previamente. Es necesario encontrar una métrica adecuada para medir objetivamente el desempeño del modelo.

6. Ajuste de parámetros: Una vez que se evaluó el modelo, posiblemente se quieran mejorar los resultados, por lo que se procede a ajustar los hiper-parámetros de dicho modelo. Estos hiper-parámetros que se modifican en esta etapa no son los pesos del modelo (tal como la pendiente de la recta), sino que son más inherentes al modelo en sí. Algunos hiper-parámetros pueden ser: número de etapas de entrenamiento, tasa de aprendizaje, valores iniciales, entre otros.

7. Predicciones: Por último se realizan predicciones, que pueden contrastarse con un conjunto de datos que el modelo no haya visto nunca, que suele denominarse *test set*, y se separó inicialmente del set original. La idea de este set no es modificar el modelo, sino realizar una estimación de su error de generalización. Esto es, saber con que precisión dicho modelo predice valores no vistos.

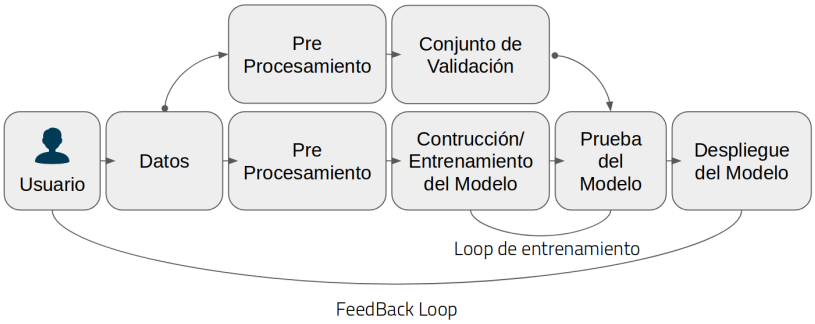


Figura 1.3: Estructura utilizada para el entrenamiento.

Capítulo 2

Paradigmas del aprendizaje

"Hay alguien tan inteligente que aprende de la experiencia de los demás"
— Voltaire, 1694-1778

2.1. Overfitting

Overfitting es el fenómeno en donde, aunque se siga mejorando el ajuste a los datos de entrenamiento, no se consigue mejorar el error de generalización o predicciones sobre datos no vistos anteriormente.

El error de generalización es una medición de qué tan preciso es el algoritmo prediciendo valores nunca vistos. En términos estadísticos, aunque es útil realizar una estimación del error en el sub-conjunto de entrenamiento, este no incluye todos los miembros de la población y por lo tanto, inferencias a partir del mismo generalmente difieren de las características de la población completa.

Overfitting resulta de ajustar los datos más de lo necesario. Es el caso en que el error entre el modelo propuesto y los datos de entrenamiento ya no es un buen indicativo de qué tan bien ajustará a nuevos datos. Este error entre modelo y datos de entrenamiento se denomina *in-sample error*. El error entre la función real y el modelo propuesto se denomina error de generalización o *out-of-sample error*. *Overfitting* se presenta cuando se disminuye el *in-sample error* pero aumenta el *out-of-sample error*.

En la figura 2.1 se muestran tres imágenes que esquematizan los distintos casos de error de generalización. Se tiene una función original a partir de la cual se obtienen algunos puntos, y se les agrega un ruido *gaussiano*. Luego se realizan ajustes con polinomios de distintos órdenes: 1, 4 y 15. Arriba de cada gráfico se muestra el error de generalización o *out-of-sample error* (MSE_original); y el error del ajuste o *in-sample error* (MSE_muestras). Se ve que el *in-sample error* disminuye a medida que aumenta

la complejidad del modelo, es decir, que pasa más cerca de cada uno de los puntos de entrenamiento. Sin embargo, también se ve que el modelo se aleja de la función original. Se podría llegar al extremo de ajustar un polinomio de grado n-1, con n el número de puntos, obteniéndose un *in-sample error* igual a cero dado que pasaría por cada punto y, sin embargo estaría muy alejado de la función que se quiere aproximar.

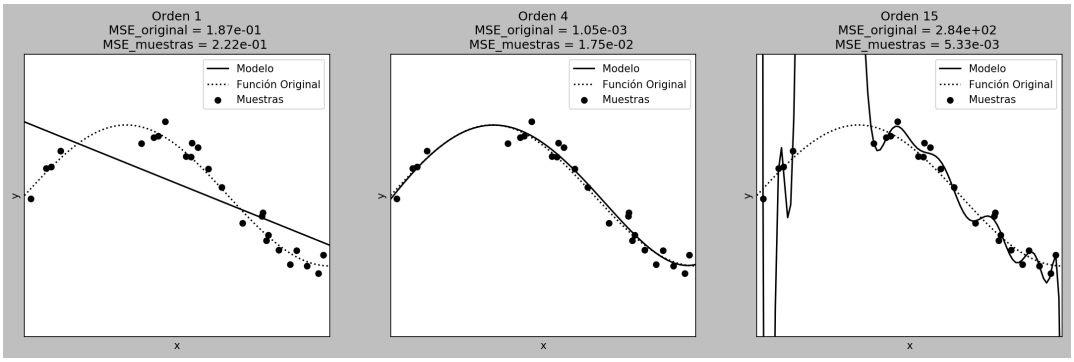


Figura 2.1: Estructura utilizada para el entrenamiento.

Lo que el modelo ve son los datos, no la función original. Si el modelo tiene demasiados parámetros libres, comenzará a ajustar el ruido de los datos. Tanto mayor será el *Overfitting* mientras mayor sea el nivel de ruido. Por otro lado disminuirá con el aumento de la cantidad de datos.

Además, mientras mayor sea la complejidad de la función objetivo, se tendrá mayor tendencia a realizar *Overfitting*. La intuición de esto es que dado el modelo que mejor ajusta la función objetivo, la parte que no esta siendo modelada adecuadamente actúa como ruido en los datos. Esto se conoce como ruido determinístico y se esquematiza en la figura 2.2, donde la complejidad de f aparece como ruido para el modelo $\boldsymbol{h}^*$. Un breve resumen de los parámetros que influyen en este fenómeno se presentan en la tabla 2.1.

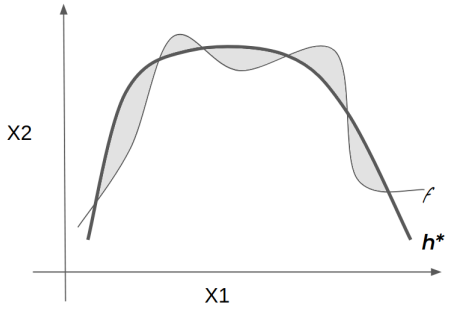


Figura 2.2: Ruido Determinista.

Número de Puntos	↑	Overfitting	↓
Ruido	↑	Overfitting	↑
Complejidad de la función objetivo	↑	Overfitting	↑

Tabla 2.1: Resumen de los parámetros que contribuyen en el *Overfitting*.

2.1.1. Regularización

La regularización es una de las primeras herramientas con las que se cuenta para combatir el *Overfitting*. Consiste en aplicar una restricción al algoritmo de aprendizaje para mejorar el *out-of-sample error*, especialmente cuando hay gran cantidad de ruido presente en los datos.

La regularización es tanto un arte como una ciencia. Aunque muchos métodos tienen una buena base matemática, gran cantidad de los utilizados en la práctica son heurísticos. Una de las formas de ver la regularización es como una penalidad Ω a la complejidad del modelo:

$$E_{out}(h) \leq E_{in}(h) + \Omega(H), \tag{2.1}$$

donde h es el modelo o hipótesis elegida entre el conjunto completo de hipótesis H. Es decir que h puede ser $2x + 1$ del conjunto completo H de los polinomios de primer orden. $E_{out}(h)$ es el *out-of-sample error*, $E_{in}(h)$ es el *in-sample error*.

La ecuación 2.1 muestra que aunque se elija un polinomio que pase por cada punto de los datos y el *in-sample error* sea cero, se tendrá una gran penalización por utilizar un polinomio tan complejo y, efectivamente, el *out-of-sample error* aumentará. Ω puede ser un vector de valores que acompañen al vector de coeficientes de un polinomio, penalizando los coeficientes asociados a los grados más altos del polinomio. De esta manera se favorece polinomios de primer orden y por lo tanto más simples.

2.1.2. Validación

Otra herramienta para minimizar el *Overfitting* es la validación. En ambas herramientas se busca minimizar E_{out}, sin embargo, en esta última, en vez de ponerle una restricción al valor del mismo, se trata de estimarlo.

Lo que se busca es separar un conjunto de datos llamado conjunto de validación. Se entrena el modelo y luego se utiliza el conjunto de validación para realizar una estimación del error de generalización, y con esto realizar ciertas decisiones en el proceso de aprendizaje. Dado que el modelo nunca vio los datos de validación, se tendrá una estimación imparcial del *out-of-sample error*.

El principal uso que tiene la validación es la selección de modelos. Puede utilizarse para elegir el orden del polinomio con el cual ajustar los parámetros de regularización, y cualquier decisión que afecte el proceso de aprendizaje. Debe tenerse en cuenta que una

vez utilizado el conjunto de validación, por ejemplo, para la selección de un modelo, éste habrá sido utilizado en el proceso de aprendizaje y si se continua empleando, arrojará una estimación sesgada de E_{out}. Esto se debe a que *se seleccionó* el modelo con el menor error de validación, y por lo tanto este tendrá un sesgo optimista.

La cantidad de datos que se separan para conformar el conjunto de validación no es trivial, y en gran cantidad de casos la decisión se basa en medidas empíricas. Supongamos que K es el número de datos que se deja para el conjunto de validación, y N el número de datos totales. Se quiere saber como generaliza un modelo entrenado con los N datos. Mientras más chico sea K, menor será la discrepancia entre el modelo entrenado con (N-K) datos, y el modelo ideal entrenado con N datos. Mientras K sea mayor, mejor será la estimación del error de generalización del modelo con el que se entrenó, es decir del modelo (N-K). Esto es, mientras mayor sea K, mejor será la estimación del error de generalización de un modelo que no es de interés, pero mientras menor sea K, peor será la estimación sobre el modelo que se quiere. Si se contara con infinitos datos esto no sería un problema, dado que se tomaría un K suficientemente grande para realizar una buena estimación y aún así N-K sería un número suficientemente grande. En la práctica difícilmente se cuente con datos de sobra. Esto lleva a que la elección de K sea compleja, y se utilice el método de *cross-validation*, método que se esquematiza en la figura 2.3.

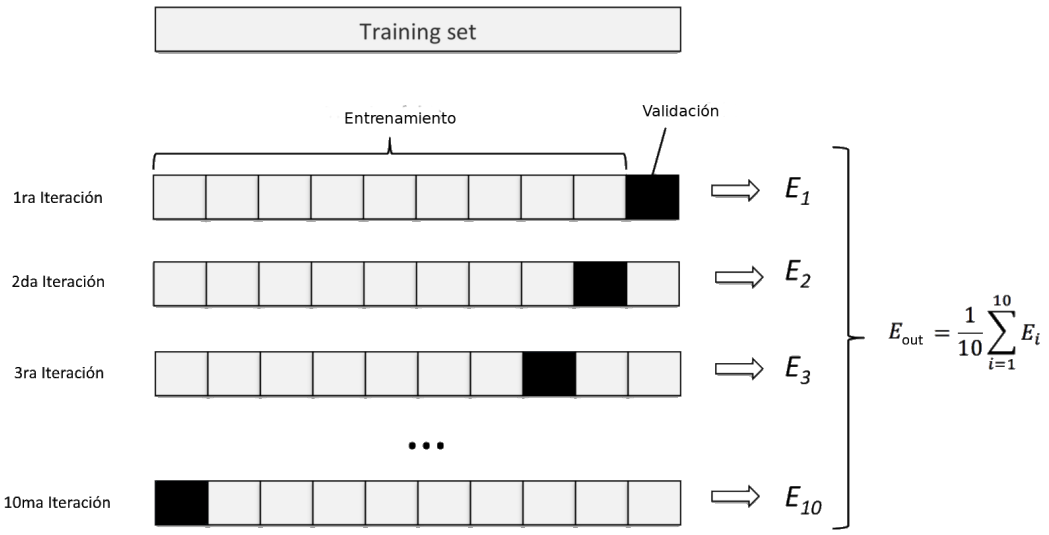


Figura 2.3: *Cross-Validation*, herramienta contra *Overfitting*.

El método de *cross-validation* consiste en separar un conjunto de datos para validación, entrenar y luego estimar el *out-of-sample error* con dicho conjunto de validación. Esto es repetido con distintos sub-conjuntos de validación y luego se promedian todas las estimaciones. De este modo se aprovechan la totalidad de los datos para entrenamiento, y a su vez se tiene una buena estimación del error de generalización.

2.2. Los tres principios del aprendizaje

El hecho de hacer que una computadora «aprenda» es un concepto complejo y amplio. Aprender de los datos tiene algunos principios inherentes que, aunque son simples, no son menores. Pasar por alto estos principios en el entrenamiento trae como consecuencia predicciones que se alejan de la realidad sin haber levantado alarmas en el camino. Es fundamental entenderlos y comprender sus limitaciones.[2]

2.2.1. La navaja de Occam

La navaja de Occam se relaciona con la elección del modelo. Este principio se atribuye a William de Occam (1287-1347), donde la «navaja» hace referencia a recortar el modelo hasta que quede lo más simple posible. El principio dice que «una solución simple es más probable que sea correcta que una compleja.»

Si las posibilidades de que algo ocurra son altas, entonces si el hecho ocurre, éste no es significativo. En cambio si las posibilidades de que ocurran son bajas, el hecho de que ocurra es significativo. Esto se refleja con hipótesis o modelos más complejos o simples. La posibilidad de que un modelo complejo ajuste a los datos son altas, por lo que si lo hace, no tiene mayor significancia. Mientras que el modelo más simple que se logre encontrar y que explique los datos, tendrá una mayor significancia estadística.

El modelo más simple que ajuste los datos es también el más plausible. Cuando se habla de un modelo simple se hace referencia a los grados de libertad del modelo. En los polinomios esto es simple de ver, a mayor grado del polinomio mayor es la complejidad del modelo. Sin embargo, la cantidad de parámetros que tenga un modelo no es directamente sus grados de libertad. En muchos modelos los parámetros no son independientes entre sí, y la dimensión efectiva es menor que la cantidad de parámetros. Encontrar dicha dimensión efectiva no suele ser tan fácil y, usualmente, se recurre a heurística más que un número exacto.

Existe controversia en si la navaja de Occam es un principio inherente a la naturaleza. Aunque en la ciencia de datos se lo toma como tal, debe hacerse la aclaración de que no hay gran cantidad de evidencia empírica o fundamento matemático que pruebe que el principio es válido para cualquier ámbito del universo. Haciendo esa salvedad, en ciencia de datos se lo utiliza para tomar el modelo más simple que sea útil para cumplir el propósito, aunque esto implique un modelo que ajuste pobremente a los datos. La razón de esto es que el precio que se paga por la complejidad del modelo puede ser demasiado en comparación al beneficio del propio ajuste.

2.2.2. Sampling Bias

Sampling Bias hace referencia a la parcialidad de los datos con los que se entrene, y por lo tanto la importancia de una recolección de datos adecuada. El principio marca que si los datos son obtenidos con un sesgo, el aprendizaje va a retornar un resultado sesgado. Resulta simple e intuitivo, pero es incontable la cantidad de veces que se pasa por alto este principio de forma inadvertida a la hora de elegir los datos de entrenamiento. En muchos casos, aunque se es consciente del principio, no puede hacerse demasiado para evitarlo. Si se ignora este principio se obtendrán resultados sesgados, en muchos casos demasiado optimistas, y no se tendrá forma de comprobarlo. *Sampling Bias* se refiere a la forma en la que se toman los datos, más que los datos en sí.

Idealmente, cuando se realiza investigación, se deben seleccionar los datos muestrales de forma totalmente aleatoria de la población en estudio. Cuando el investigador falla en seleccionar datos aleatorios, corre el riesgo de impactar la validez de los resultados dado que la muestra no representa fielmente la población de interés.

Un ejemplo muy simple sería hacer una encuesta online sobre el uso de la computadora. Claramente arrojará resultados sesgados, dado que las personas que lo hagan tendrán un interés marcado hacia la tecnología.

Un ejemplo famoso sobre *Sampling Bias* ocurrió en la Segunda Guerra Mundial, en donde el estadístico Abraham Wald analizó la distribución de los disparos que habían sufrido los aviones por parte de anti-aéreos. Estudiando dónde recibían la mayor cantidad de disparos, se pondría protección extra en los aviones. Con la distribución dada según la figura 2.4, se concluyó que debían reforzarse las alas y el cuerpo del avión. Esta fue una conclusión errónea que surgió de solo tener en cuenta los aviones que *sí* retornaban a la base. Aquellos que estaban siendo dañados en la cabina, motores y partes de la cola, no sobrevivían la misión. Por lo tanto, lo que realmente había que reforzar eran dichas partes.[4]

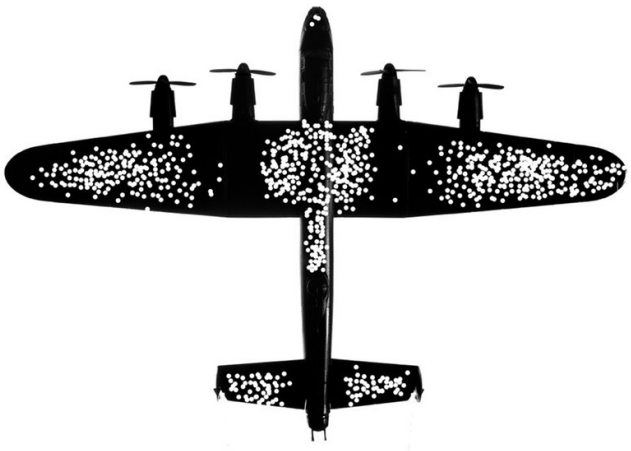


Figura 2.4: Ejemplo de *Sampling Bias*. Ejemplo de los aviones de Abraham Wald.

2.2.3. Data Snooping

El principio subyacente propone que si los datos han sido afectados, en cualquier etapa de aprendizaje, la habilidad del modelo para predecir correctamente ha sido comprometida. Cuando se realiza *Data Snooping*, pueden encontrarse patrones en los datos que parecen significativos, cuando en realidad no hay una verdadera relación de trasfondo. Esto ocurre por la reutilización de los datos. Si se trata de entrenar un modelo, y luego otro, y así sucesivamente, eventualmente se tendrá éxito. En otras palabras, tras probar muchos modelos sólo se le presta atención al que arrojó resultados positivos.

Para saber que tanto afectó el hecho de haber realizado *Data Snooping*, se debe tener en cuenta la penalidad por la complejidad del modelo. La complejidad del modelo no solo es la del modelo que arrojó resultados positivos, dado que de los otros modelos también se «aprendió», y debe tenerse en cuenta que el aprendizaje conllevó un modelo efectivo más complejo.

Un ejemplo simple de *Data Snooping* es la normalización previa a la separación de los conjuntos de entrenamiento y validación. Normalizar antes de la separación de los datos es un error, dado que se contamina el conjunto de validación que el modelo no debería ver solo hasta el final. Información del conjunto de validación aparece en los datos de entrenamiento como una normalización, y los resultados o predicciones tendrán un sesgo optimista cuando se realice la validación.

Las formas de lidiar con este fenómeno son: una disciplina estricta en el manejo de los datos, teniendo precaución de no contaminar los mismos. Además, debe tenerse en cuenta que si se realizó *Data Snooping*, se mantenga un registro de qué tanto se están contaminando los datos, y tratar la precisión de los resultados con precaución dada esta contaminación.

Capítulo 3

Pre-Procesamiento

"Excelente maestro es aquel que, enseñando poco, hace nacer en el alumno un deseo grande de aprender"
— Arturo Graf, 1848-1913

3.1. Introducción a los datos

Se trabajó con datos de telemetría provenientes de una plataforma del área aeroespacial provista por la empresa INVAP. La telemetría tiene una frecuencia de sampleo de 32 segundos. Los datos provienen de sensores físicos, tales como temperaturas, voltajes, corrientes, así como también provienen de acciones de control y datos de validación.

En la figura 3.1 se muestran algunas de las variables con las que se trabaja. Las mismas se encuentran normalizadas según la media y el desvío estándar. Puede observarse periodicidad en algunas variables, así como también variables que se comportan como la función escalón y representan una variedad de datos de validación. Puede verse que también se encuentran variables con gran cantidad de ruido. Además es notable la falta de continuidad en varias secciones de todas las telemetrías, esto es un efecto de la forma de adquisición de los datos.

3.1.1. Datos Faltantes

Como ya se mencionó, toda la telemetría se solicita a la plataforma con una frecuencia de 32 segundos. Sin embargo, no hay una sincronización en el envío de los mismos. Esto provoca que para un dado punto en el tiempo se cuenta con unas variables y otras no, de manera que cuando se guardan los datos habrá muchos puntos temporales para los cuales no se reportan valores. La base de datos con la que se trabaja rellena estos datos faltantes con el tipo de datos *NaN* (*Not a Number*).

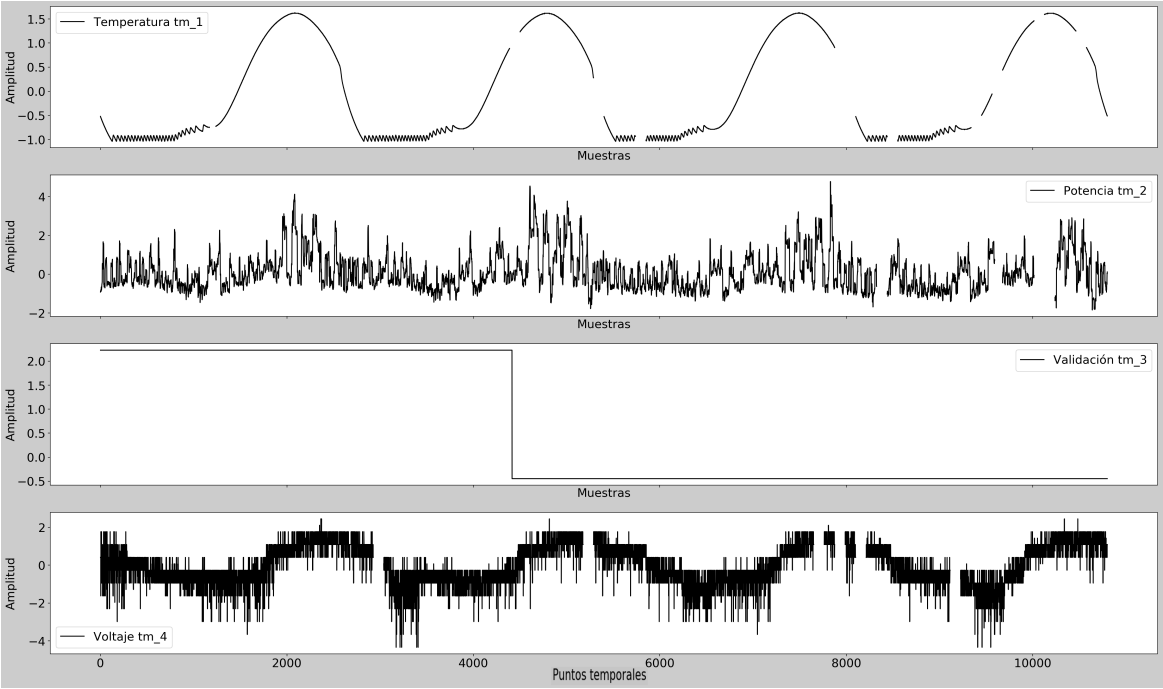


Figura 3.1: Ejemplo de los datos con los que se trabaja. Datos periódicos, validaciones y mediciones ruidosas. Se muestra una porción pequeña de los datos.

Se tienen distintas formas de tratar esto. Cada forma puede repercutir en los resultados finales, y tiene ventajas y desventajas. La forma más conservadora de solucionar el problema es descartando todas las filas, es decir todos los puntos temporales, que tengan al menos un valor *NaN*. Esto tiene el beneficio de no generar nuevos datos artificiales, sin embargo, se pierde gran cantidad de datos y los que quedan tienen discontinuidades notables. Computacionalmente este método no tiene mayor complejidad o exigencia.

La segunda forma es rellenar los valores faltantes con el dato de la columna más cercana. De esta forma se conservan todos los datos, pero se crean datos artificiales y siguen habiendo discontinuidades. Aunque el método no es exigente en sí, cuando se lo hace a millones de datos puede ser intensivo en la memoria.

La tercera forma, y la utilizada a lo largo del trabajo, es realizar una interpolación en las columnas sobre los datos inexistentes. Se comprobó que para todos los fines del trabajo, alcanza con una interpolación lineal, y no es necesario una de mayor orden. Como se mencionó anteriormente, los métodos no son exigentes computacionalmente en sí mismos, pero dado que se trabaja con millones de datos, se vuelven notoriamente demandantes de memoria. Este método genera datos artificiales, pero se entiende que continúan el patrón de las telemetrías y afectan positivamente el resto de los algoritmos.

En la figura 3.2 se representan esquemáticamente los métodos mencionados.

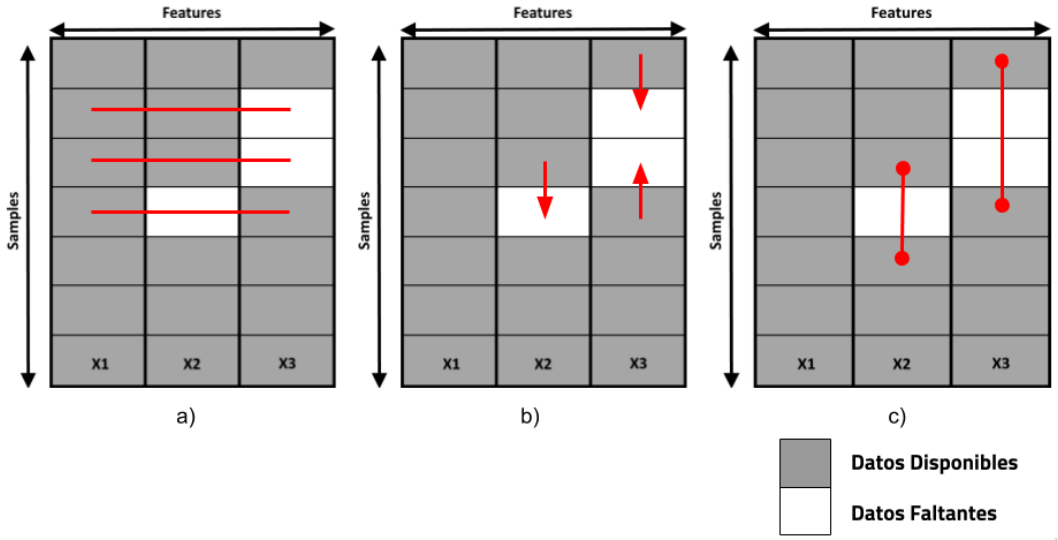


Figura 3.2: Distintas formas de tratar los NaN's. a) Eliminar las filas. b) Rellenar con el más cercano. c) Interpolar.

En la figura 3.3 se aplican los métodos mencionados a la primer variable, *Temperatura tm_1*, presentada en la figura 3.1. Se observa que cuando se eliminan las filas que contienen NaN's, se cambia esencialmente el patrón de los datos, lo cual perjudica a los algoritmos enfocados en la detección de cambios en dichos patrones.

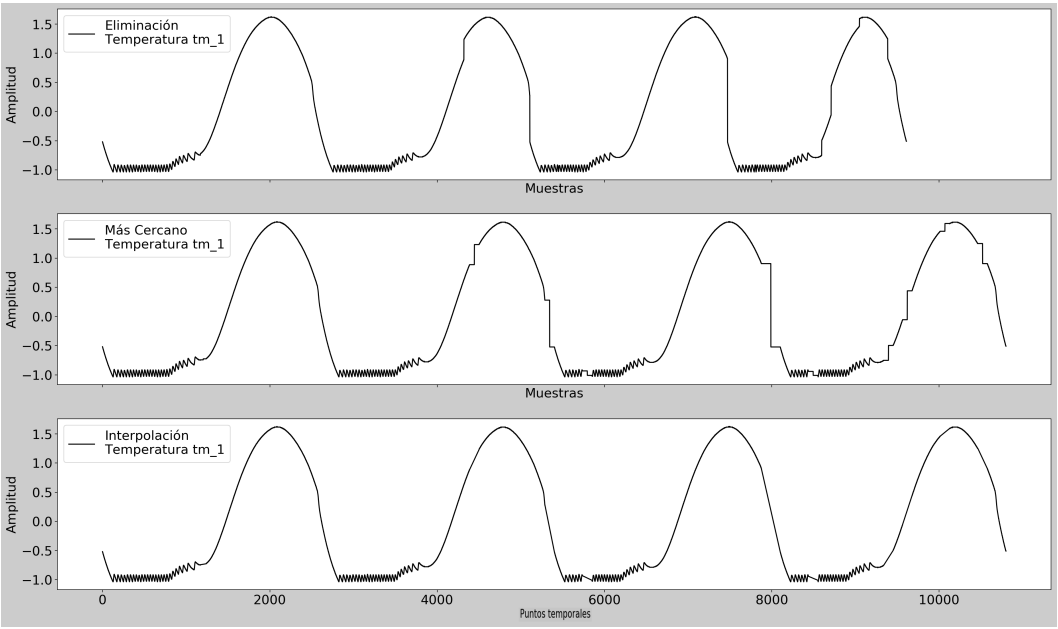


Figura 3.3: Los distintos métodos para tratar el faltante de datos, aplicados a una variable.

3.1.2. Normalización

Los valores y escalas de las diferentes variables pueden diferir enormemente. Por ejemplo, pueden haber mediciones de voltaje en milivoltios o voltios. Por lo tanto,

un cambio de escala o normalización suele ser, aunque no siempre, necesaria. Existen muchos algoritmos que son sensibles al intervalo de tiempo y escala de los datos. El intervalo de tiempo de los datos esta normalizado desde la adquisición de los datos.

La forma más simple de normalización es llevando los datos a una escala de [0,1]. Se utiliza el máximo y mínimo de cada columna para obtener la normalización. Se realiza según la ecuación:

$$Z = \frac{X - min}{max - min} \tag{3.1}$$

Esto tiene como desventaja una distorsión indeseada en la distribución de los datos. El costo de tener un rango acotado es que se termina con desviaciones estándar más pequeñas y hay una tendencia a la supresión de puntos aislados.

Una normalización o estandarización más aceptada es la *Normalización Z*, la cual se da según la siguiente ecuación:

$$Z = \frac{x - \bar{X}}{\sigma} \tag{3.2}$$

en donde se computa la media $\bar{X}$ y el desvío estándar σ de cada columna o variable. Esto lleva a que la variable tenga una media cero y un desvío estándar unitario y sin embargo no tenga su rango de valores limitados. Dado el objetivo de este trabajo, es fundamental no suprimir puntos aislados y es por ello que se escogió esta estandarización.

3.2. Resampleo

Se puede cambiar la frecuencia temporal de los datos dependiendo del modelo a usar. Por ejemplo, en el caso de una predicción mediante extrapolación de alguna variable, tanto mejor será la predicción mientras más datos se tengan. En contraste, en el método de *Clusters*, no resulta tan necesario contar con millones de datos. Por limitaciones de Hardware, se trata de reducir los datos temporales tratando de no afectar el desempeño de los modelos.

Se agrupan filas, o datos temporales, y se les aplica alguna función que los lleve a una sola fila. Cuántas filas y qué función aplicar dependen del conjunto de datos. Por experiencia previa, se decidió utilizar la mediana como la función a aplicar. Para elegir cuántas filas, se realizó un análisis de Fourier sobre todas las variables. Esto mostró que la mayoría de las variables periódicas presentaban un periodo igual o mayor a 24 horas.

En este trabajo se aplicó un resampleo para usarse con el algoritmo de *Clusters*, y se mostrarán resultados para distintos tamaños de reampleo en el siguiente capítulo.

3.3. Dimensionalidad

3.3.1. La maldición de la dimensión

La Maldición de la dimensión, o mejor conocida en inglés como *Curse of Dimensionality*[5] se refiere a una variedad de serios desafíos que aparecen cuando se trabaja con datos de gran cantidad de dimensiones, y es un factor importante en el diseño de los algoritmos de *Machine Learning*. Este término fue utilizado por Bellman (1961), en el contexto de la *Teoría de la aproximación*, para enfatizar el hecho de que la dificultad de generar buenas estimaciones no solo crece con la dimensión (lo cuál no es novedad) sino que lo hace exponencialmente.

El punto en común de los problemas que surgen es que, a medida que la dimensionalidad crece, el *volumen* del espacio crece exponencialmente más rápido y los datos disponibles se vuelven dispersos. En la figura 3.4 puede verse que el número de puntos necesario para mantener la distancia promedio entre los puntos, crece exponencialmente con el número de dimensiones.[1]

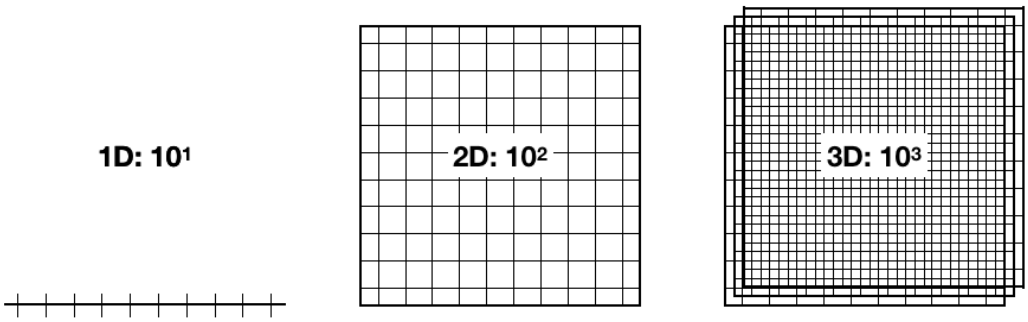


Figura 3.4: El número de datos necesarios para mantener la distancia promedio constante.[1]

Es de particular interés entender esta problemática, dado que afecta fuertemente a la medición de distancias entre puntos, una de las técnicas del método de *Clusters*. El problema de la alta dimensionalidad tiene una base muy general, y es que nuestra intuición proviene de tres dimensiones. Por ejemplo, la mayor parte del volumen de una hiper-esfera, es decir una esfera de más de tres dimensiones, se encuentra en su cáscara. Otro ejemplo es para el caso de Gaussianas multidimensionales donde la mayoría de los puntos no se encuentran cerca de su media, sino que más cerca de la campana multidimensional.

La conclusión de esto es que muchas veces no es útil tener más variables. Si algunas variables no aportan nada, como puede ser un sensor redundado para la medición de temperatura, resulta conveniente deshacerse de dicha variable y dejar sólo una de ellas.

Pueden utilizarse algoritmos para reducción de dimensionalidad que favorecen a los modelos de aprendizaje. Usualmente se puede dividir en dos categorías:

* Selección de *features* o características: Se descartan las variables innecesarias quedándose solo con las más relevantes. Para decidir qué variables descartar pueden utilizarse distintos criterios o métricas.

* Reducción de dimensionalidad: Se trata de encontrar un espacio vectorial más pequeño que el inicial. Este nuevo espacio que se encuentra resulta de una combinación lineal de las variables originales, e idealmente contiene la misma información.

Entre los métodos de Selección de *features* puede utilizarse un filtro de baja varianza. Este filtro descarta las variables con varianza nula o muy baja, dado que una variable con valores constantes no aporta al aprendizaje del modelo. Otro método es un filtro de alta correlación, el cuál analiza la correlación entre las variables y descarta aquellas que estén por encima de algún umbral. Si dos variables tienen tendencias similares, es probable que sólo puedan aportar la misma información, lo que puede disminuir el desempeño del modelo.

Por otro lado, el método más utilizado en reducción de dimensionalidad es el *Análisis de Componentes Principales* o PCA, por sus siglas en inglés. Este método se presenta en el siguiente capítulo.

Capítulo 4

Modelos

"Todo hombre es superior a mi en algún sentido. En ese sentido, aprendo de él"
— Ralph Waldo Emerson, 1803-1882

4.1. Principal Component Classifier (PCC)

Se utilizó el método de detección de anomalías basado en componentes principales y detección de puntos aislados, o *outliers*, propuesto por Mei-Ling Shyu (*et. al.*) en [6]. Esta metodología, conocida como *Principal Component Classifier*, tiene el beneficio de que sólo algunos de los componentes principales deben guardarse luego del entrenamiento para realizar las predicciones, por lo cual el cálculo puede realizarse en muy poco tiempo y utilizarse en detección en tiempo real. Antes de presentar el método es necesario estudiar qué es el Análisis de Componentes Principales.

4.1.1. Principal Component Classifier (PCA)

El Análisis de Componentes Principales[7] se utiliza para reducción de dimensionalidad, con el propósito de un análisis más simple de los datos, pudiendo visualizar los mismos en 2D o 3D; así como también para evitar la *maldición de la dimensionalidad*. Es un método simple y elegante que trata de explicar la estructura de varianza-covarianza de los datos, haciendo una transformación lineal a un nuevo espacio vectorial.

Esta transformación lineal tiene tres propiedades fundamentales: (1)los componentes principales no están correlacionados, (2) el primer componente principal tiene la varianza más alta, el segundo componente principal tiene la segunda varianza más alta, y así sucesivamente; y (3) la variación total en todos los componentes principales combinados es igual a la variación total en las variables originales. Este nuevo espacio

vectorial se obtiene fácilmente de los autovectores de la matriz de covarianza o de la matriz de correlación.

Obtener los componentes principales de una u otra matriz no es indiferente. Usualmente no se obtienen los mismos autovectores, ni es fácil hallar la relación entre ellos. En general, se utiliza la matriz de correlación para realizar el cálculo dado que no es susceptible a la escala de los datos. Esto no es tan relevante si se han estandarizado los datos.

Se plantea un conjunto de datos que consiste en m vectores,

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m\}, \tag{4.1}$$

donde cada uno está compuesto por n elementos,

$$\boldsymbol{x}_i^T = \{x_{i,1}, x_{i,2}, ..., x_{i,n}\}. \tag{4.2}$$

Saber qué significa cada vector o elemento depende de cada caso, o de cada conjunto de datos, en particular. Por ejemplo, cada vector podría representar un píxel de una imagen, y cada elemento los valores RGB y de intensidad. En este trabajo muchos de los vectores representan mediciones, y cada elemento, el valor de una de las n variables medidas.

La covarianza es una medida de la dependencia lineal entre dos variables aleatorias, matemáticamente definida como:

$$cov(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle(\boldsymbol{x}_i - \langle\boldsymbol{x}_i\rangle)(\boldsymbol{x}_j - \langle\boldsymbol{x}_j\rangle)\rangle \tag{4.3}$$

donde $\langle ... \rangle$ es la esperanza o media del vector. De forma análoga, $\boldsymbol{x}_i$ y $\boldsymbol{x}_j$ se calculan según:

$$cov(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{m}\sum_{k=1}^{m} x_{i,k}x_{j,k} - \mu_i\mu_j, \tag{4.4}$$

donde μ_i yμ_j es la esperanza o media de cada vector.

La covarianza cuantifica la dependencia entre dos variables. Un valor positivo indica que el aumento de una de las variables corresponde con el aumento de la otra variable. Análogamente, con un valor negativo. Da una medida de qué tan en conjunto se mueven ambas variables. Un valor cero, indica que ambas variables son no tienen una dependencia lineal.

Es fundamental notar que la covarianza es una medida de la dependencia lineal entre las variables. Es decir que si dos variables se relacionan de forma no lineal, el valor de la covarianza será bajo. Esto es una desventaja en la metodología de PCA, aunque este problema puede ser subsanado utilizando lo que se conoce como *kernels*, según sea necesario.

Una vez definida la covarianza, la correlación se define sencillamente según:

$$corr(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{cov(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sigma_i \sigma_j}, \tag{4.5}$$

donde σ_i y σ_j son los desvíos estándar de cada vector.

Cuando sólo se tienen dos variables, se dispone de fórmulas explícitas; cuando se tiene un conjunto de datos de más dimensiones, puede definirse una matriz de correlación o covarianza considerando cada par de componentes:

$$C_{i,j} = cov(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{4.6}$$

$$\boldsymbol{C} = \begin{bmatrix} cov(\boldsymbol{x}_1, \boldsymbol{x}_1) & cov(\boldsymbol{x}_1, \boldsymbol{x}_2) & \dots & cov(\boldsymbol{x}_1, \boldsymbol{x}_m) \\ cov(\boldsymbol{x}_2, \boldsymbol{x}_1) & cov(\boldsymbol{x}_2, \boldsymbol{x}_2) & \dots & cov(\boldsymbol{x}_2, \boldsymbol{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ cov(\boldsymbol{x}_m, \boldsymbol{x}_1) & cov(\boldsymbol{x}_m, \boldsymbol{x}_2) & \dots & cov(\boldsymbol{x}_m, \boldsymbol{x}_m) \end{bmatrix}$$

Debe notarse que por definición la matriz de covarianza es simétrica, por lo que está asegurada la existencia de m autovectores independientes. En otras palabras, existe $\boldsymbol{V}$ tal que:

$$\boldsymbol{C} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T \tag{4.7}$$

con $\boldsymbol{V}^T = \boldsymbol{V}^{-1}$y donde $\boldsymbol{D}$ es la matriz llena de ceros salvo en su diagonal (donde se colocan los autovalores de $\boldsymbol{C}$). $\boldsymbol{V}$ es la matriz con los autovectores de $\boldsymbol{C}$, ordenados por columnas.

Un ejemplo simple de los autovectores de la matriz de correlación para datos generados por una gaussiana 2D se puede ver la figura 4.1. En dicha figura se ve que el Componente Principal 'PC0', está en la dirección que mayor variación tienen los datos y corresponde al autovector de mayor autovalor.

El método de PCA consiste en proyectar todos los datos en estos nuevos vectores. Teniendo en cuenta que cada medición ya se estandarizó con la media y el desvío estándar, cada nueva variable puede escribirse según:

$$\boldsymbol{y}_j = \boldsymbol{V}\boldsymbol{x}_j, \tag{4.8}$$

lo cual es una proyección de los datos en los componentes principales. Como se mencionó anteriormente, luego de proyectar al nuevo espacio vectorial, la correlación entre las nuevas variables es nula, como se puede ver en la figura 4.2 con los datos del ejemplo anterior.

Además, dado los autovalores de la matriz de correlación: $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_m \geq 0$, la variable $\boldsymbol{y}_j$ tiene varianza λ_j; y la propiedad mencionada anteriormente, donde la

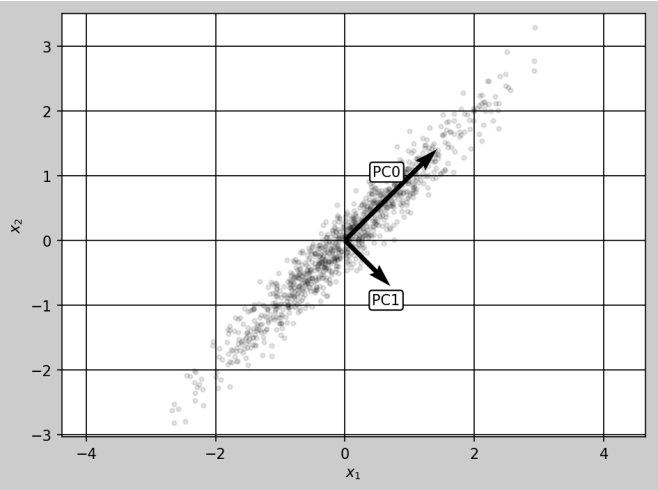


Figura 4.1: Datos provenientes de una Gaussiana 2D. Ilustración de los Componentes Principales: 'PC0' y 'PC1'.

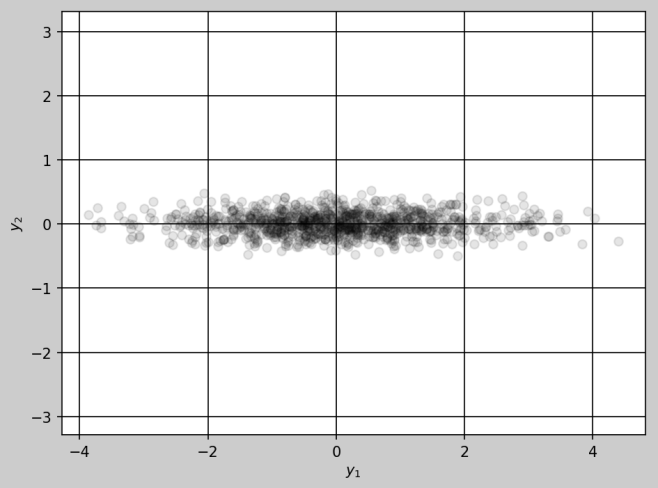


Figura 4.2: Ilustración de la proyección en los componentes principales de los datos del ejemplo anterior.

suma de la varianza del nuevo espacio es igual a la varianza total del espacio de entrada:

$$\sum_{i=0}^{m} var(\boldsymbol{x}_i) = \sum_{i=0}^{m} \lambda_i, \tag{4.9}$$

donde $var(\boldsymbol{x}_i)$ es la varianza de $\boldsymbol{x}_i$ y puede definirse como $cov(\boldsymbol{x}_j, \boldsymbol{x}_j)$. Dado que la mayor variación queda en el primer componente principal, se puede descartar la segunda variable, y quedarse con un espacio de una sola dimensión, aún con la mayoría de la variación del sistema inicial. En un caso de más dimensiones, con cuántas variables o con qué porcentaje de la varianza quedarse no es es obvio y debe escogerse algún criterio para ello.

Se mencionó que el método de PCA puede utilizarse para una mejor visualización integral de los datos. Aplicando este método al conjunto de datos de trabajo de este

proyecto se obtiene la figura 4.3 para los primeros dos componentes principales y la figura 4.4 para los primeros tres componentes principales. Para el conjunto de datos

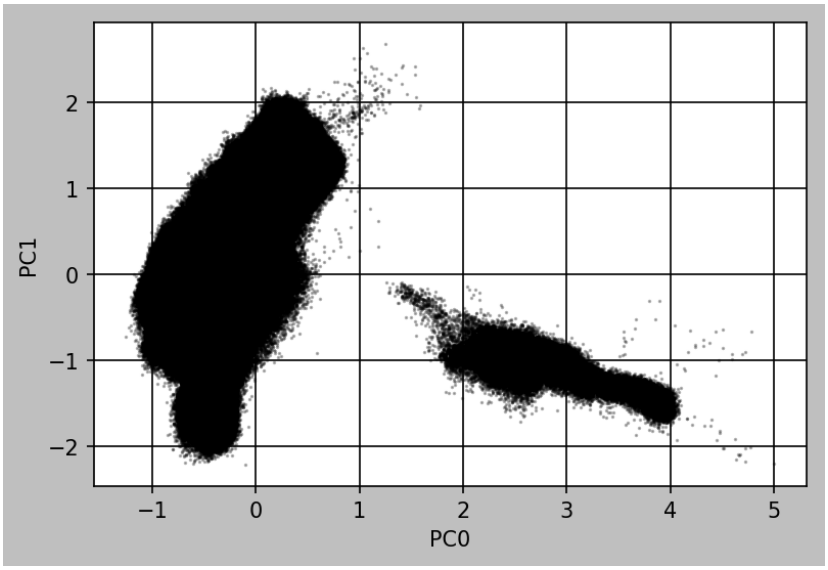


Figura 4.3: Primeros dos componentes principales de los datos de trabajo. Representan el 36.4 % de la varianza original.

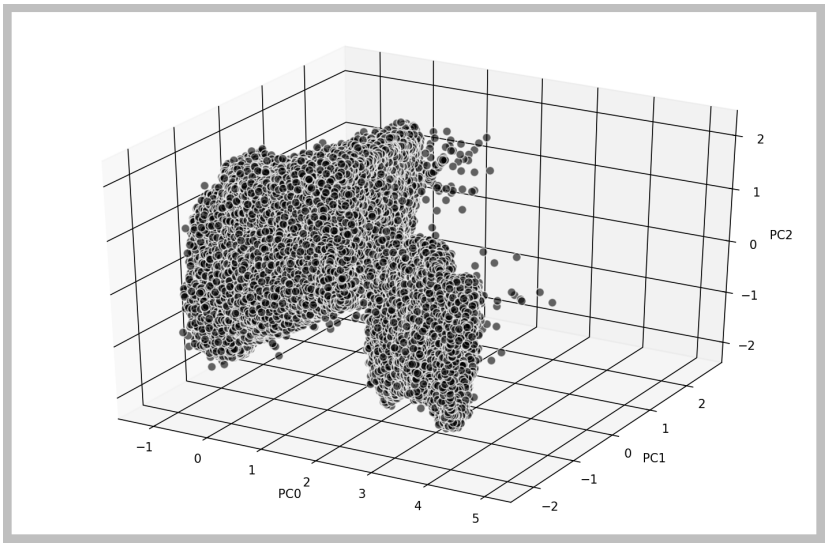


Figura 4.4: Primeros tres componentes principales de los datos de trabajo. Representan el 46.3 % de la varianza original.

de trabajo, los primeros dos componentes principales representan el 36.4 % de la varianza original, mientras que los primeros tres suman un total del 46.3 % de la varianza original.

Aunque no es fácilmente extrapolable a mayor cantidad de dimensiones, es notable lo agrupados que se encuentran los datos. Esto se debe principalmente a la normalización realizada. Datos con una alta densidad espacial son susceptibles a ser clasificados o agrupados en *Clusters*, algoritmo que se presenta más adelante.

Debe aclararse que para las figuras 4.3 y 4.4 se estandarizaron los datos, tal cual se requiere para la aplicación del método PCA, con un fin ilustrativo. Sin embargo para continuar con el desarrollo del método de PCC (Principal Component Classifier) se continúa con los datos originales.

4.1.2. Distancia Mahalanobis

Muchas técnicas de detección de anomalías se basan en el concepto de distancia. El objetivo es saber qué tan cerca se encuentran los datos a analizar, y eventualmente agruparlos en *Clusters*. Sin embargo, que dos puntos estén cerca o lejos depende qué métrica se utilice para realizar la medición. La distancia más intuitiva es la Euclideana, que dados dos vectores $\boldsymbol{x}$ e $\boldsymbol{y}$, se define como:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^T (\boldsymbol{x} - \boldsymbol{y})}. \tag{4.10}$$

Sin embargo, esta distancia tiene la desventaja de que todas las variables contribuyen por igual, lo cual suele ser poco beneficioso en algunas aplicaciones. En casos donde las variables tienen escalas diferentes entre sí, las distancias calculadas estarán dominadas sólo por algunas variables cuyas escalas sean mayores. Pueden estandarizarse las variables para tratar de subsanar este problema; sin embargo, sigue existiendo la *Maldición de la Dimensionalidad*, y variables que en las proyecciones 2D o 3D se ven cercanas pueden en realidad tener una gran distancia en mayores dimensiones.

Una alternativa a esta distancia, que tiene en cuenta la variabilidad de las mediciones, es la distancia de Mahalanobis[8], la cual se define como:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^T C^{-1} (\boldsymbol{x} - \boldsymbol{y})}, \tag{4.11}$$

donde $\boldsymbol{C}$ es la matriz de correlación de todo el conjunto de datos. Esta distancia puede interpretarse como una idea multi-dimensional generalizada de medir a cuántos desvíos estándar se encuentran los puntos a la media de la distribución. La distancia de Mahalanobis se refiere a la distancia relativa de los puntos al *centroide*. El *centroide* es un punto multi-dimensional donde las medias de cada variable se intersectan.

Cuando se analizan variables por sí mismas, pueden tratar de detectarse anomalías observando los puntos que son muy grandes o muy pequeños en relación al resto. Sin embargo, cuando se analizan conjuntos de variables la situación es más compleja. En gran cantidad de dimensiones pueden aparecer *outliers* que no sean puntos extremos en ninguna variable por separado y, por lo tanto, pasen desapercibidos en la detección. Es por ello que es importante considerar todas las variables a la hora de detectar anomalías. La distancia de Mahalanobis cumple este rol de analizar que tan extremo es un punto teniendo en cuenta la media multi-dimensional.

Suponiendo un conjunto de datos que consiste en m vectores,

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m\}, \tag{4.12}$$

donde cada uno está compuesto por n elementos,

$$\boldsymbol{x}_i^T = \{x_{1,i}, x_{2,i}, ..., x_{n,i}\}, \tag{4.13}$$

que pueden escribirse en forma matricial $(n \times m)$ donde cada columna es un vector de mediciones de $(n \times 1)$:

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n1} & \dots & x_{nm} \end{bmatrix},$$

donde se define:

$$\boldsymbol{x}_i' = [x_{i,1}, x_{i,2}, ..., x_{i,m}], \tag{4.14}$$

como el vector de observaciones en las variables $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m$, para un dado punto en el tiempo.

Transformando al nuevo espacio formado por los autovectores de la matriz de correlación según:

$$\boldsymbol{y}_j = \boldsymbol{V}\boldsymbol{x}_j, \tag{4.15}$$

se obtiene la matriz:

$$\boldsymbol{Y} = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1m} \\ y_{21} & y_{22} & \dots & y_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n1} & \dots & y_{nm} \end{bmatrix},$$

donde la suma de los cuadrados de los elementos del componente principal estandarizados asociado a $\boldsymbol{x_j'}$,

$$\sum_{i=1}^{m} \frac{y_{j,i}^2}{\lambda_i}, \tag{4.16}$$

es equivalente a la distancia de Mahalanobis de la observación $\boldsymbol{x_j'}$ a la media del conjunto.

Además, dado que los componentes principales no están correlacionados, suponiendo distribución normal de los datos iniciales y asumiendo que la cantidad de datos es

grande resulta que la sumatoria:

$$\sum_{i=1}^{q} \frac{y_{j,i}^2}{\lambda_i} = \frac{y_{j,1}^2}{\lambda_1} + \frac{y_{j,2}^2}{\lambda_2} + ... + \frac{y_{j,q}^2}{\lambda_q} \qquad q \leq m, \tag{4.17}$$

tiene una distribución Chi-Cuadrado con q grados de libertad. Los primeros componentes principales representan una gran porción de la varianza de los datos originales. Estos componentes tienden a estar fuertemente relacionados con las variables originales que tienen grandes varianzas o covarianzas. Por lo tanto, las anomalías que se encuentren en los primeros componentes representan puntos anómalos, posiblemente extremos, en una o más de las variables originales. Es decir, se comporta como una anomalía puntual o contextual.

Por otro lado, puede hacerse el mismo análisis que en 4.17, pero para los últimos r componentes

$$\sum_{i=m-r+1}^{m} \frac{y_{j,i}^2}{\lambda_i} \tag{4.18}$$

Los últimos componentes principales son más sensibles a mediciones que son inconsistentes con la estructura de los datos, y que sin embargo son anómalos con respecto a la variable individual.

En la figura 4.5 el punto A corresponde a una medición que es una anomalía puntual tanto en la variable X_1 como en la variable X_2, pero no es una anomalía en cuanto a la estructura de los datos. Esta anomalía aparecería observando los primeros componentes principales. Por otro lado, el punto B no es un valor extremo en ninguna de las dos variables; sin embargo, no pertenece a la estructura de los datos y es más susceptible a ser detectada con los últimos componentes principales.[9]

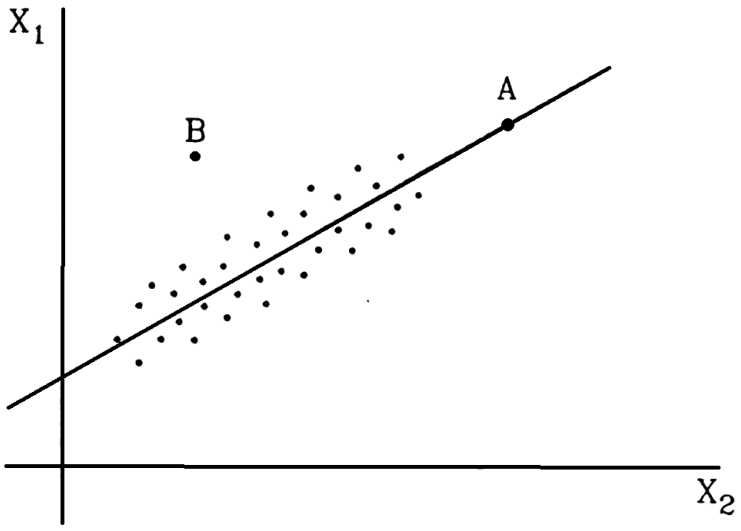


Figura 4.5: Los dos tipos de anomlías detectables con PCC.

4.1.3. Estimador de la matriz de correlación

Es importante que los datos de entrenamiento estén libres de datos anómalos, dado que estos pueden traer aparejado un incremento en las varianzas, covarianzas y correlaciones. Variaciones en estos últimos parámetros trae consecuencias en la matriz de correlación y por lo tanto en el cálculo de los componentes principales. Por ello es importante encontrar un estimador robusto de la matriz de correlación. Un inconveniente a tener en cuenta es que usar estimadores robustos de la matriz de correlación no garantiza que la nueva matriz encontrada sea definida positiva, y por lo tanto puede ser necesario verificar que los nuevos autovalores encontrados sean efectivamente mayor a cero.

El método más utilizado (y el empleado en este trabajo) para obtener un estimador robusto de la matriz de correlación y el vector de medias o promedios es llamado *multivariate trimming*. Ésta utiliza la métrica de Mahalanobis para identificar un porcentaje arbitrario α de los valores extremos en las mediciones a ser descartados o recortados (*trimmed*). Es un método iterativo donde en cada iteración se descarta el $\alpha\,\%$ de los valores extremos. Se comienza por calcular los parámetros muestrales, el vector de promedios $\boldsymbol{\mu}$ y la matriz de correlación $\boldsymbol{C}$. Con ello se calcula la distancia de Mahalanobis como se mostró en 4.11 para todos los vectores $\boldsymbol{x}'$. Se encuentra el $\alpha\,\%$ de los puntos que pertenecen a los máximos valores y se los descarta. Con los datos restantes se vuelven a calcular el vector de promedios $\boldsymbol{\mu}$ y matriz de correlación $\boldsymbol{C}$. Mientras el número de observaciones sea mayor que el número de *features*, el estimador de la matriz de correlación va a ser definido positivo. En otras palabras, mientras el tamaño de un dado $\boldsymbol{x}'_j$ sea menor que el tamaño de $\boldsymbol{x}_j$. Por cada iteración del método se obtiene un estimador que aproxima mejor a la variable poblacional. En la figura 4.6 se muestran las distancias de Mahalanobis de los datos de trabajo antes y después de aplicar *multivariate trimming*. El valor de α se seleccionó igual a 0,002 según recomendaciones de [6].

4.1.4. Implementación

Con lo analizado, el método de PCC consiste en clasificar el punto $\boldsymbol{x}'_j$ como anómalo si se cumple que:

$$\sum_{i=1}^{q} \frac{y_{j,i}^2}{\lambda_i} > C_1 \qquad \text{o} \qquad \sum_{i=m-r+1}^{m} \frac{y_{j,i}^2}{\lambda_i} > C_2, \tag{4.19}$$

y se clasifica como nominal si cumple:

$$\sum_{i=1}^{q} \frac{y_{j,i}^2}{\lambda_i} \leq C_1 \qquad \text{y} \qquad \sum_{i=m-r+1}^{m} \frac{y_{j,i}^2}{\lambda_i} \leq C_2, \tag{4.20}$$

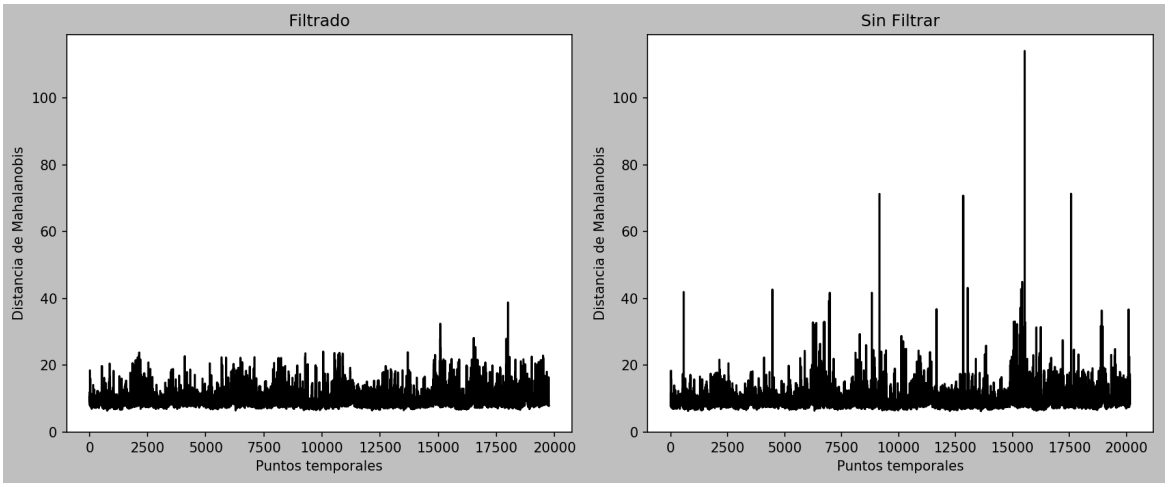


Figura 4.6: Distancia de Mahalanobis para los datos resampleados cada una hora. Con y sin filtro del estimador de la matriz de correlación.

donde C_1 y C_2 son los umbrales tales que el método produzca un determinado porcentaje de falsas alarmas. Como se mencionó anteriormente, dada una distribución normal de los datos, se espera que las sumatorias tengan una distribución Chi-Cuadrado. Sin embargo, para obtener un valor más acorde a los datos de trabajo, y desligarse de la hipótesis de distribución normal, se realizan los histogramas de las sumatorias y se hallan C_1 y C_2 tal que se acumule un porcentaje determinado. Esto se muestra para los datos del proyecto en la figura 4.7

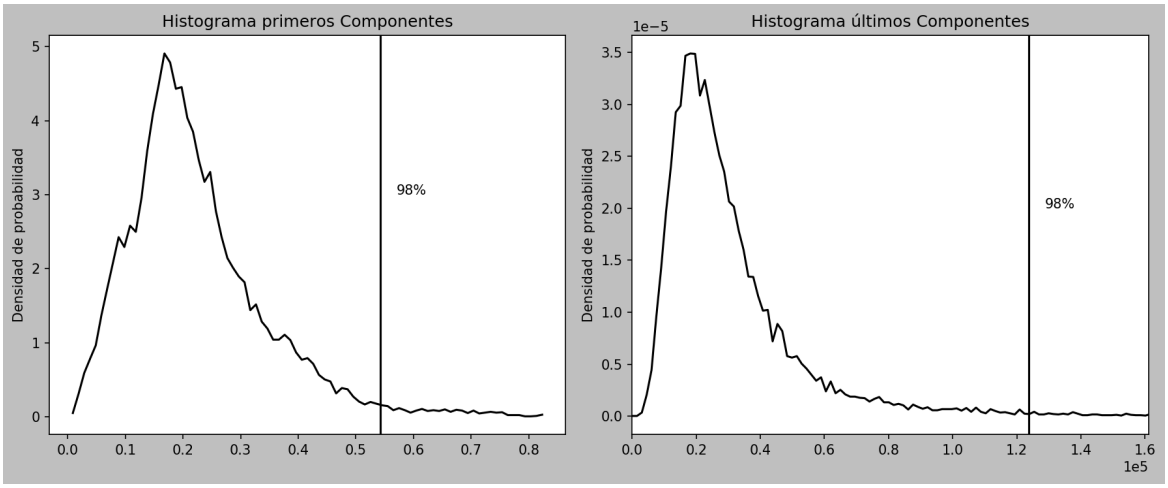


Figura 4.7: Histograma de las sumas de los cuadrados de los elementos de los componentes principales estandarizados.

4.2. Clustering

Clustering es una metodología en donde se agrupan conjuntos de datos en alguna forma tal que los datos pertenecientes al mismo grupo sean más similares entre si, en

algún sentido, que a los datos pertenecientes al resto de los grupos. Este análisis es el más conocido cuando se refiere a aprendizaje no supervisado. El mismo asigna etiquetas a datos no clasificados y es de gran utilidad dado que identifica estructura en conjuntos de datos de los cuales no se sabe a que clase pertenecen. Este es el caso con el que se trabaja, donde no se cuenta con datos etiquetados en *anómalos* o *nominales*. Debe tenerse en cuenta que las clasificaciones que se encuentren no tienen porque seguir una estructura obvia y, posiblemente, no una esperada.

Clustering es una metodología más que un algoritmo. Lo que diferencia a los distintos algoritmos dentro de este análisis es qué se entiende por el concepto de *Cluster* o grupo. Pueden agruparse los datos según alguna métrica que defina la distancias entre los puntos, según la densidad de puntos que hay en una determinada región del espacio, según la distribución estadística que tengan los datos y, básicamente, cualquier parámetro que pueda ser útil para relacionar distintos grupos de datos.

4.2.1. Estacionalidad

Antes de comenzar el análisis de *Clustering* sobre los datos en sí, se realizó un análisis de frecuencia de los mismos, donde se encontró una gran estacionalidad en los datos. Es decir que las distintas variables tienen un comportamiento muy marcado en las distintas estaciones del año.

La forma en que se analizó la estacionalidad fue la siguiente: para una dada variable $\boldsymbol{x}_j$ se buscó en la matriz de correlación las variables que se correlacionaran con ella a partir de un umbral dado. Se utilizaron estas variables como los términos independientes para realizar un ajuste lineal a la variable $\boldsymbol{x}_j$. Se realizaron ajustes con períodos de un día a lo largo del período de estudio. A los pesos resultantes del ajuste se les realizó un análisis de frecuencia con Fourier.

Se realiza el ajuste según la siguiente fórmula:

$$\boldsymbol{x}_j = \sum_{i=0}^{k} w_i \boldsymbol{x}_i, \tag{4.21}$$

donde $\boldsymbol{x}_j$ es la variable objetivo y las $\boldsymbol{x}_i$ salen del análisis de correlación. Se encuentran los w_i que minimicen el error cuadrático medio.

Debe aclararse que el propósito de los ajustes lineales es sólo el análisis de frecuencias. Éstos modelos no tienen ninguna significancia estadística ya que se incumple una hipótesis básica, y es que las variables con las que se ajusta $\boldsymbol{x}_i$ deben ser independientes entre sí. Estas variables fueron sacadas según qué tan bien correlacionadas estén con la variable objetivo $\boldsymbol{x}_j$ y, por lo tanto, tienen un mayor o menor grado de dependencia lineal entre sí.

En la figura 4.8 se muestra cómo varían los distintos pesos que se usaron para

ajustar una variable de temperatura arbitraria del conjunto de datos de trabajo. Se observa un período marcado anual, y transformando a frecuencia, se encuentra que los siguientes períodos fundamentales son seis y tres meses. El período de tres meses tiene un fuerte comportamiento estacional, por lo que se decidió realizar un análisis de cada estación del año por separado.

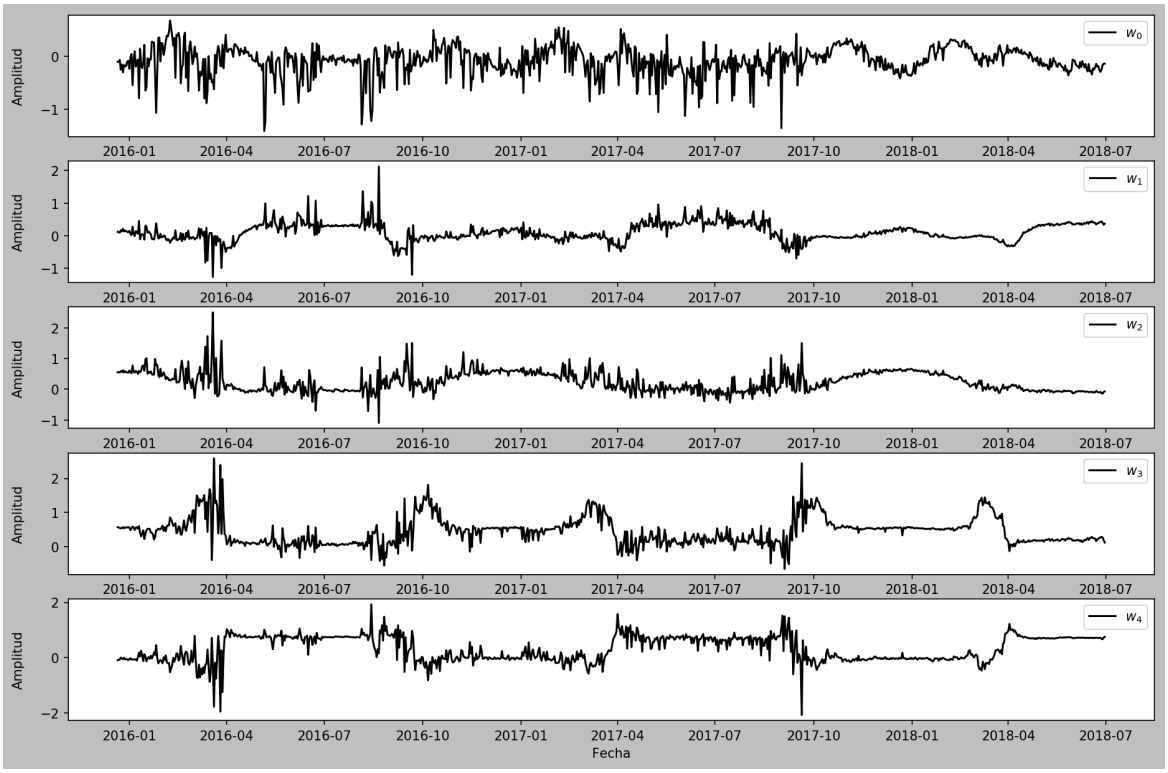


Figura 4.8: Se muestra cómo varían los distintos pesos que se usaron para ajustar una variable de temperatura arbitraria del conjunto de datos de trabajo. R^2 promedio para todos los días igual a 0,994.

Para realizar el análisis por estaciones se dividió el período de estudio en bloques de 91 días, centrando los equinoccios y solsticios en cada período de análisis. Con los datos ya normalizados, resampleados cada tres horas y con los filtros mencionados, se procedió a realizar una transformación PCA. Se encontraron los componentes principales asociados al primer bloque de 91 días, y se proyectó el resto de los bloques sólo a esos vectores. Se encontró empíricamente que esto daba mejores resultados que realizar una transformación PCA a cada bloque de 91 días. Proyectando a los vectores del primer bloque de días se encontró que los cúmulos eran más densos y se diferenciaban mejor entre sí. Estos resultados se muestran en la figura 4.9 para los primeros dos componentes principales. Puede apreciarse que hay cúmulos muy marcados que son susceptibles a ser agrupados en *clusters* según una métrica de distancia estándar como la Euclideana. Además, dada la distribución de los datos, puede asociarse cada *cluster* a una distribución normal o Gaussiana de dos dimensiones.

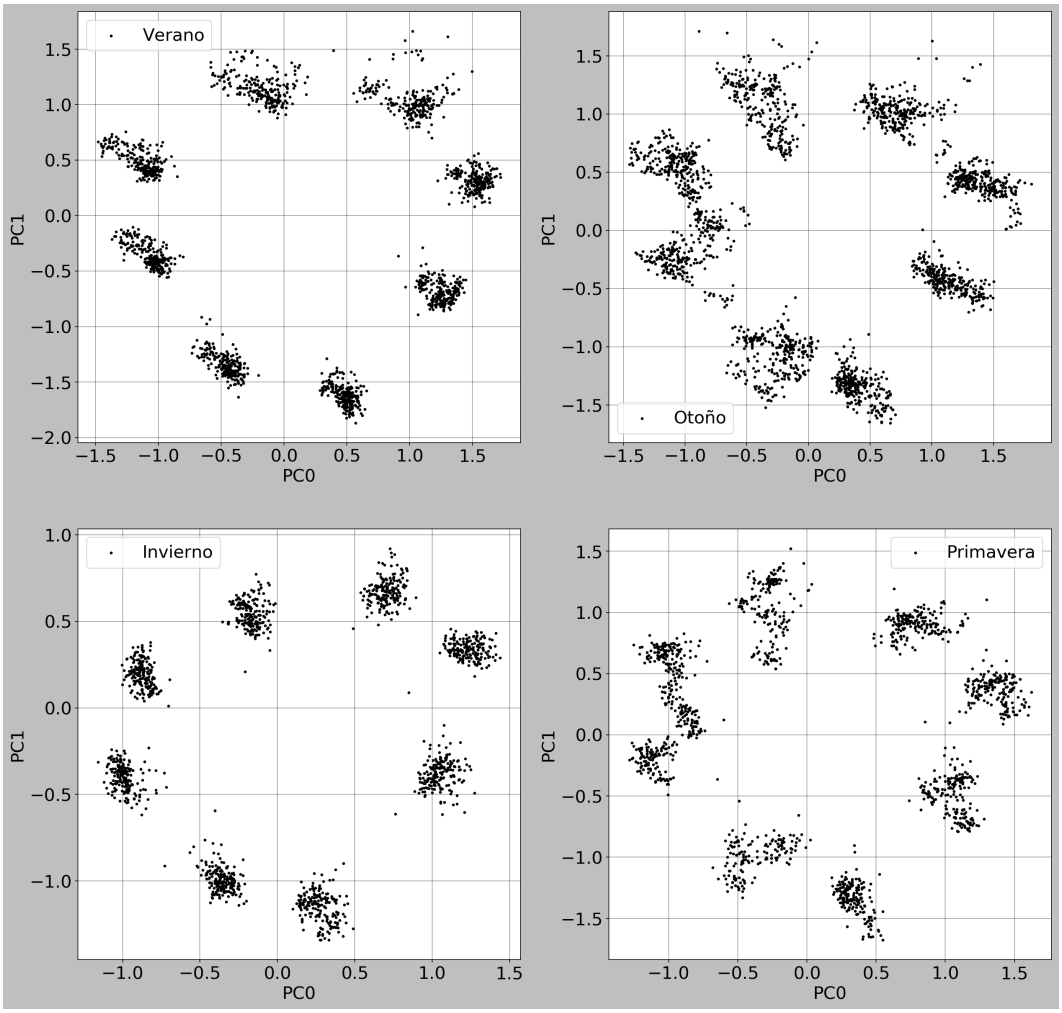


Figura 4.9: Transformación PCA para los primeros dos componentes principales para las cuatro estaciones del año.

4.2.2. K-Means

El algoritmo *K-Means*[7] es de los más utilizados en *clustering*, principalmente dada su simpleza. El objetivo del algoritmo es encontrar grupos en los datos, donde el número de grupos a encontrar es un parámetro definido. Se trabaja de forma iterativa para asignar cada punto a uno de estos grupos basado en alguna métrica suministrada, usualmente distancia Euclideana. *K-Means* devuelve como resultado los centroides de los grupos encontrados y las etiquetas de los datos, según a qué grupo pertenecen.

El método busca minimizar la siguiente función:

$$J(\boldsymbol{V}) = \sum_{i=0}^{c} \sum_{j=0}^{c_i} \|\boldsymbol{x}'_j - \boldsymbol{v}_i\|^2, \tag{4.22}$$

donde c_i es el número de puntos pertenecientes al i-ésimo *cluster*, c es el número de *clusters*, y $\boldsymbol{V} = \{\boldsymbol{v_1}, \boldsymbol{v}_2, ..., \boldsymbol{v}_c\}$ son los centroides.

Las etapas del algoritmo son las siguientes:

1. Se seleccionan c *clusters* aleatoriamente.

2. Asignar cada punto de los datos al centroide más cercano.

3. Volver a calcular el centro de cada *cluster* como la media de los puntos pertenecientes a dicho *cluster*:
$$\boldsymbol{v}_i = \frac{1}{c_i} \sum_{j=0}^{c_i} \boldsymbol{x}'_j \tag{4.23}$$

4. Reasignar cada punto al centroide más cercano.

Se itera sobre los pasos hasta que no haya reasignación de puntos o que los centroides se desplacen menos que un umbral establecido.

Silhouette Score

Para saber cuántos *clusters* realizar se buscó alguna métrica que midiera qué tan bien agrupados estaban los datos dado que se los agrupaba de alguna manera específica. Para ello se utilizó la puntuación Silhouette[7], la cual mide qué tanto pertenecen los puntos a su *cluster* comparado a otros grupos. La misma toma un valor entre -1 y 1. Un valor más cercano a uno representa una buena configuración de *clusters*.

Suponiendo que los *clusters* ya se crearon y por lo tanto cada dato tiene su etiqueta, se define con una métrica Euclideana:

$$a(\boldsymbol{x}'_i) = \frac{1}{c_i - 1} \sum_{j \in C_i, i \neq j} \|\boldsymbol{x}'_j - \boldsymbol{x}'_i\|, \tag{4.24}$$

$$b(\boldsymbol{x}'_i) = \min_{i \neq j} \frac{1}{c_j} \sum_{j \in C_j} \|\boldsymbol{x}'_j - \boldsymbol{x}'_i\|, \tag{4.25}$$

con lo que se tiene el valor *Silhouette* para un dado punto según:

$$s(\boldsymbol{x}'_i) = \frac{b(\boldsymbol{x}'_i) - a(\boldsymbol{x}'_i)}{max\{a(\boldsymbol{x}'_i), b(\boldsymbol{x}'_i)\}} \tag{4.26}$$

donde C_i es el i-ésimo *cluster*. Puede interpretarse $a(\boldsymbol{x}'_i)$ como una medida de qué tan bien asignada está la etiqueta del punto $\boldsymbol{x}'_i$; mientras menor el valor, mayor su pertenencia al *cluster*.

$b(\boldsymbol{x}'_i)$ es la menor distancia promedio de $\boldsymbol{x}'_i$ a todos los puntos pertenecientes a algún *cluster* del cual no sea miembro. El *cluster* que cumpla esto se lo llama '*cluster* vecino' y es el siguiente mejor *cluster* para el punto $\boldsymbol{x}'_i$. Mientas mayor sea $b(\boldsymbol{x}'_i)$ menor duda habrá de que fue asignado al grupo adecuado.

La idea del método es hacer un promedio de todos los $s(\boldsymbol{x}'_i)$, lo cual dará un estimado de qué tan bien están asignadas las etiquetas de todos los datos. Se utiliza algún

método de *clustering* con k grupos, donde se va calculando la Puntuación Silhouette para distintos k y se selecciona el número de *clusters* que mejor puntaje dé.

El resultado de aplicar este método a los datos de trabajo utilizando el algoritmo K-Means se muestra en la figura 4.10.

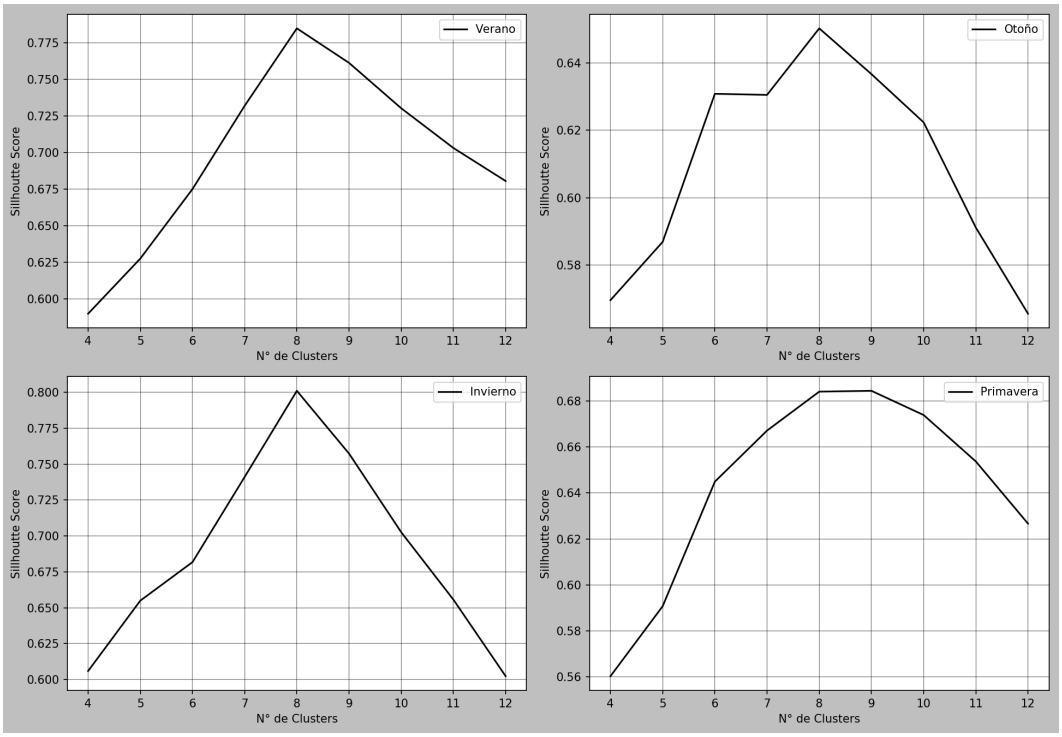


Figura 4.10: Puntuación Silhouette para las cuatro estaciones cuando se utiliza el algoritmo K-Means.

Es visible en la figura 4.9 que hay grupos marcados, y a simple vista pueden contarse ocho de ellos. Con la puntuación Silhouette puede corroborarse esto, dado que el promedio máximo arroja que una configuración de ocho *clusters* es la adecuada.

4.2.3. Gaussian Mixture Model (GMM)

Gaussian Mixture Model[7] es un modelo probabilístico que asume que todas las mediciones provienen de una distribución N-dimensional de Gaussianas con parámetros desconocidos. Este algoritmo trata de encontrar esos parámetros desconocidos. Puede pensarse en este modelo como una generalización del modelo de K-Means, el cual incorpora información de la estructura correlacional de los datos.

Se dice que las asignaciones de este método son 'blandas'. Esto quiere decir que dado un punto, se tiene con qué probabilidad pertenece a cada grupo, y no una asignación única. Esto significa que cada punto de los datos puede haber sido generado por cualquiera de las distribuciones que se encuentren con su correspondiente probabilidad. El algoritmo para encontrar los parámetros de cada distribución se conoce como *Expectation Maximization (EM)*.

Expectation Maximization Algorithm

El algoritmo EM es una forma de encontrar estimaciones de máxima verosimilitud o *likelihood* para los parámetros del modelo cuando sus datos están incompletos. La base del modelo es elegir valores aleatorios para la forma y posición de la distribución. En el caso de una Gaussiana, se escogen aleatoriamente la media y el desvío estándar. Luego se itera sobre los siguientes dos pasos hasta la convergencia:

- Paso-E: Realizar la asignación estadística de cada punto a cada distribución. Se calcula la probabilidad de que cada punto pertenezca a cada grupo o Gaussiana.
- Paso-M: Con los pesos adecuados, según con qué probabilidad cada punto pertenece a cada grupo, se recalculan los parámetros de cada distribución.

En este trabajo se utilizó el algoritmo de K-Means para generar los primeros parámetros del algoritmo, por lo que ya no son aleatorios y facilita la convergencia. Aplicando GMM a los datos pertenecientes al verano, se obtuvo la figura 4.11.

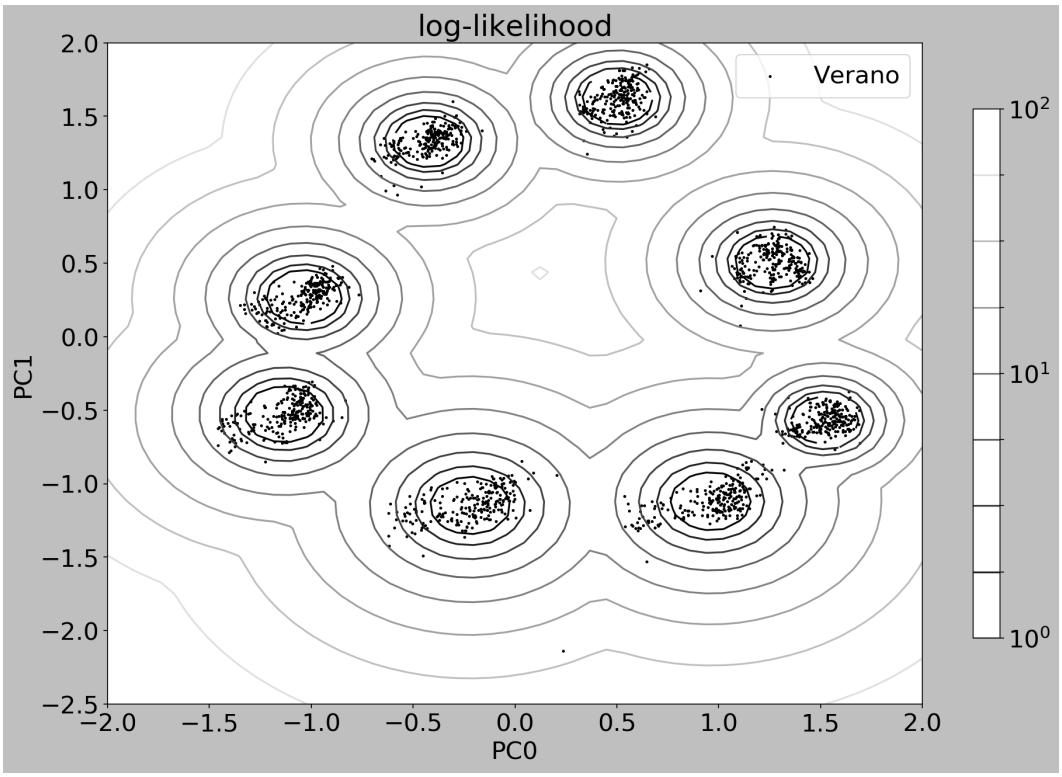


Figura 4.11: Gaussian Mixture Model aplicado a los datos con transformación PCA pertenecientes al verano.

Una vez obtenidas las distintas distribuciones pueden utilizarse los grupos para la detección de anomalías. Se toman las nuevas mediciones que se quiere saber si son nominales o no, y se calcula cuál es la función de verosimilitud o *likelihood* de que haya sido generado por algunas de las distribuciones encontradas por el método. Si el valor es menor que un umbral especificado puede tomarse ese punto como anómalo.

4.3. Forecasting

Con los dos métodos anteriores se realizó un análisis de puntos temporales multidimensionales $\boldsymbol{x}'$. Son metodologías integrales que tratan de analizar la plataforma en su conjunto y no variables por separadas. Al analizar varias variables al mismo tiempo, hay una tendencia a perder sensibilidad individual de cada *feature*. Por ello se complementa el análisis con una técnica para detectar anomalías puntuales en cada variable individual. *Forecasting*[10] permite realizar predicciones sobre cada medición, de modo que si las observaciones futuras se alejan demasiado de la predicción puede marcarse como algo no nominal y continuar un análisis más profundo.

Forecasting es un conjunto de herramientas muy amplio que tiene una gran cantidad de campos de aplicación. Pueden tratar de hacerse predicciones a largo plazo para casos económicos o predicciones de tan solo unos minutos para el caso de telecomunicaciones. Una buena predicción de los datos no se basa en aprender "de memoria" la estructura de los datos, sino de capturar patrones genuinos y relaciones que existen en los datos pasados.

En este trabajo se utilizó la herramienta de *Exponential Smoothing*[11]. Este tipo de predicción se basa en un promedio pesado de observaciones pasadas, con los pesos decayendo exponencialmente para las observaciones más antiguas. La variante más simple del método se define según:

$$\hat{x}_{t+1} = \alpha x_t + \alpha(1-\alpha)x_{t-1} + \alpha(1-\alpha)^2 x_{t-2} + ..., \tag{4.27}$$

donde α es el parámetro de *smoothing* y toma valores entre 0 y 1. Se denota $\hat{x}_t$ como una predicción y x_t como una medición para un dado t. La ecuación 4.27 puede escribirse de forma resumida según:

$$\hat{x}_{t+1} = \alpha x_t + \alpha(1-\alpha)\hat{x}_t \tag{4.28}$$

Esto requiere que haya una predicción $\hat{x}_0$ la cual no se puede calcular dado que no hay datos anteriores. Este valor se estima, junto con el valor de α, minimizando el error cuadrático (SSE):

$$SSE = \sum_{t=1}^{T} (x_t - \hat{x}_t)^2 \tag{4.29}$$

Este método es demasiado simple para analizar los datos de trabajo, dado que predice un valor constante luego del último dato observado. Por lo tanto, se utiliza la versión más compleja que tiene en cuenta que los datos pueden tener una tendencia y un término asociado a periodicidad o estacionalidad.[11]

Para esta variación del método existen dos pequeños modelos según cómo se contemplan los términos de tendencia y estacionalidad. Por un lado está el modelo aditivo

que expresa la variable temporal según:

$$\boldsymbol{x}(t) = Nivel + Tendencia + Estacionalidad + Ruido \tag{4.30}$$

Este es un modelo lineal donde los cambios a lo largo del tiempo no varían en magnitud. La tendencia es lineal en el tiempo, y por último el término de estacionalidad tiene una frecuencia y amplitud constantes.

Por otro lado está el modelo multiplicativo que expresa a las variables según:

$$\boldsymbol{x}(t) = Nivel \times Tendencia \times Estacionalidad \times Ruido, \tag{4.31}$$

donde este modelo no es lineal y los cambios a lo largo del tiempo pueden aumentar o disminuir su magnitud.

Los componentes para el modelo aditivo se escriben de la siguiente manera:

$$\begin{aligned} \hat{x}_{t+h} &= l_t + hb_t + s_{t+h-m(k+1)} \\ l_t &= \alpha(x_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \\ s_t &= \gamma(x_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}, \end{aligned} \tag{4.32}$$

donde l_t es el parámetro de $Nivel$, b_t es el parámetro de la tendencia, y s_t la componente de estacionalidad; con sus correspondientes parámetros de *smoothing*: α, β y γ. Se utiliza m para denotar la frecuencia de la estacionalidad dentro de un año. Por ejemplo, para datos con periodo mensual $m = 12$. h es el paso temporal de la predicción, es decir, los sucesivos pasos que continúan luego del último término observado. Por último se define k como la parte entera de $(h-1)/m$, lo que asegura que las estimaciones de estacionalidad provienen del último año de los datos.

De manera análoga se define el modelo multiplicativo según:

$$\begin{aligned} \hat{x}_{t+h} &= (l_t + hb_t)s_{t+h-m(k+1)} \\ l_t &= \alpha(x_t/s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \\ s_t &= \gamma\frac{x_t}{l_{t-1} + b_{t-1}} + (1-\gamma)s_{t-m} \end{aligned} \tag{4.33}$$

Aplicando descomposición STL [12], usando el lenguage R (R Core Team, 2017) [13], a los datos para diferenciar las componentes de tendencia y estacional se lo que se observa en la figura 4.12

Se utilizó el modelo aditivo 4.32 para realizar una predicción a 10 meses de una variable de la plataforma. Se le asigno un valor constante a esta variable durante

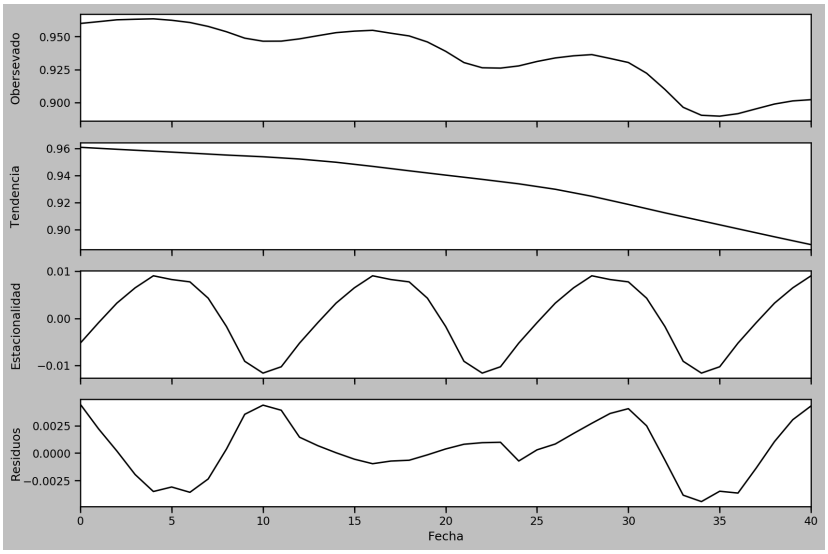


Figura 4.12: Descomposición STL para observar las componentes de tendencia, estacionalidad y el residuo de ello.

aproximadamente dos meses, simulando una situación anómala, y se quería saber si cuando volviera a sus valores nominales los valores estarían dentro de lo esperado. Caso contrario, habría ocurrido alguna anomalía mientras no se observaban los valores reales de la variable. La predicción comienza apenas la variable toma un valor constante. Esta situación puede representar el apagado del sensor por un periodo de tiempo, y se quiere saber si cuando se lo vuelva a encender los valores que presente son nominales o anómalos.

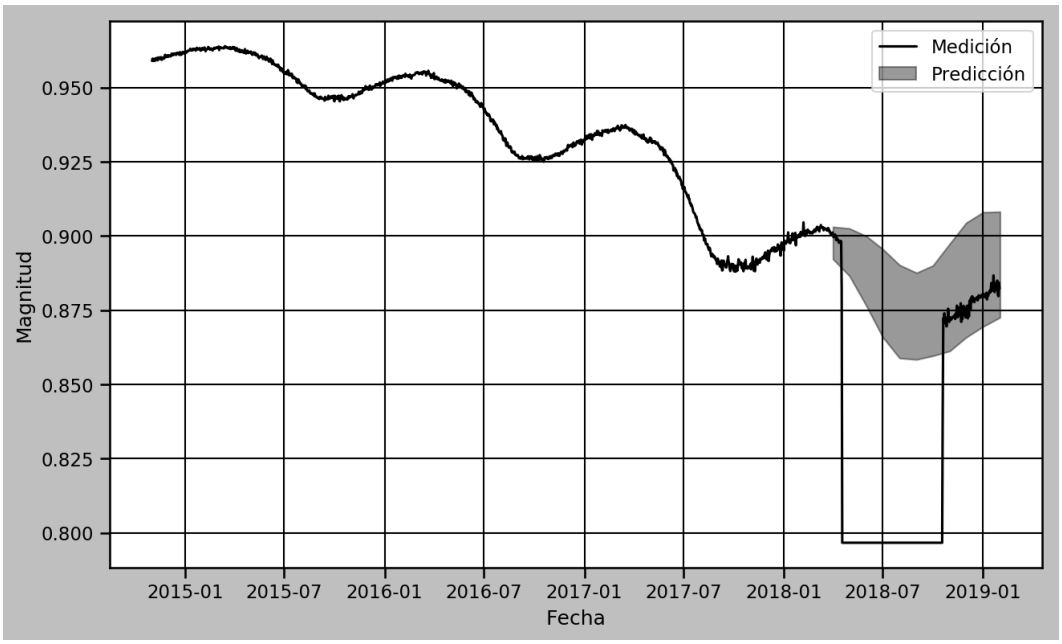


Figura 4.13: *Forecasting* de una variable desde el comienzo de la anomalía. Tras volver de la situación anómala se encuentra dentro de los valores esperados.

Se observa que cuando la variable vuelve a tener valores nominales, la misma se

encuentra dentro de la banda de tendencia. Esto marca que la variable se encuentra dentro de valores nominales. Este modelo empleado solo toma períodos de valores mensuales, por lo tanto, lo que se hizo para obtener las líneas de tendencia mínima y máxima fue resamplear los datos mensualmente tomando el mínimo y máximo de cada mes respectivamente. A esos datos resampleados se le aplicó el modelo aditivo, dado que se encontraron inestabilidades numéricas aplicando el modelo multiplicativo con estos datos específicos.

Capítulo 5

Validación

> *"Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber' "*
>
> — Albert Einstein, 1879-1955

Para probar los algoritmos planteados se procedió a realizar validaciones pertinentes. Se utilizó el conjunto ya visto, creando anomalías artificiales.

5.1. Anomalía artificial: Tendencia

Como una de las anomalías artificiales se propuso imponer una tendencia de crecimiento de temperatura sobre los sensores de dicha variable física. Se impuso un crecimiento de medio grado $°C$ por día durante un mes para un promedio total de 15 grados de aumento de temperatura. Esto puede verse en la figura 5.1. Las anomalías se colocaron de manera de abarcar una estación del año en específico.

5.1.1. Clustering

Para saber qué tan probable es que un punto nuevo haya sido generado por alguna distribución se utiliza el valor de *likelihood*. Dado un conjunto de distribuciones que pueden haber generado el nuevo dato, *likelihood* es una función que expresa qué tan probable es que el dato observado haya sido generado por la distribución en función de los parámetros de dicha distribución. Se suele denotar la función de *likelihood* según:

$$likelihood = p(D|\boldsymbol{w}) \tag{5.1}$$

donde D es el conjunto de datos de trabajo y $\boldsymbol{w}$ es el conjunto de parámetros que definen las distribuciones buscadas. Se suele entender el algoritmo de *Gaussian Mixture* como

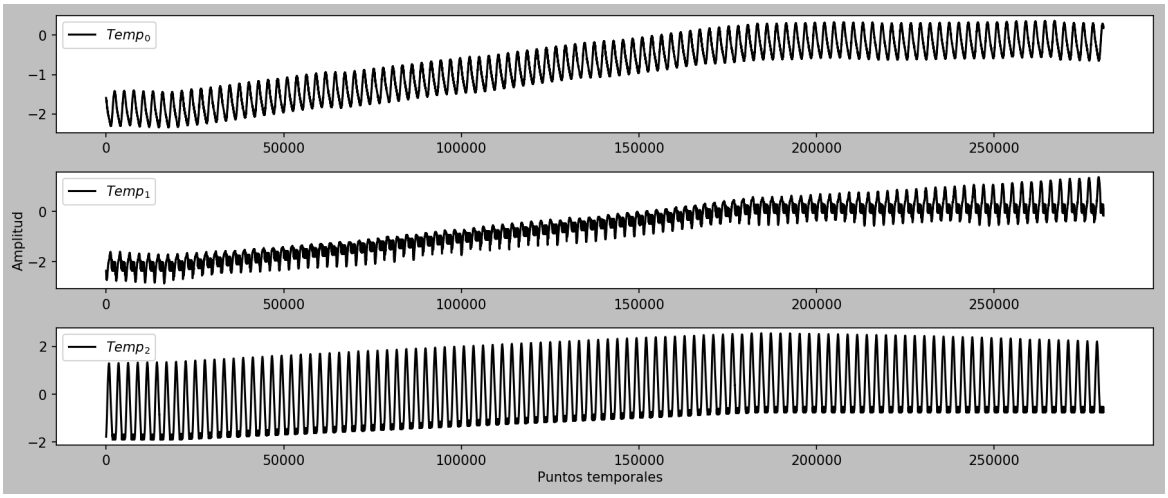


Figura 5.1: Variables de temperatura con anomalías artificiales. Valores estandarizados. Tendencia de medio grado por día por un mes.

aquel que trata de hallar $\boldsymbol{w}$ de forma de maximizar la función *likelihood* dados los datos D. En este caso, dado que ya se encontraron los parámetros $\boldsymbol{w}$ de las distribuciones, se quiere utilizar la función para saber qué tan probable es que un nuevo dato haya sido generado por cada una de dichas distribuciones.

Dado que los datos fueron transformados al espacio de los Componentes Principales, y por definición estos no están correlacionados entre sí, puede realizarse la suposición de que las dimensiones de los datos son independientes entre sí. Teniendo en cuenta esta suposición de la independencia lineal, se define a la función de *likelihood* para distribuciones gaussianas según:

$$p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\sigma}^2) = \prod_{n=0}^{N} \mathcal{N}(x_n, \mu_i, \sigma_i^2) \qquad i = 0, 1, ..., M \tag{5.2}$$

donde $\boldsymbol{x}$ es el conjunto de puntos, $\boldsymbol{\mu}$ y $\boldsymbol{\sigma}$ son el vector de medias y el vector de varianzas respectivamente de las distribuciones gaussianas $\mathcal{N}$. N es el número de datos y M es el número de distribuciones o *Clusters* en este caso.

Además,es usual utilizar *log-likelihood* que, como su nombre lo indica, consiste en tomar el logaritmo natural de la función *likelihood*. Esto hace que la multiplicación de distribuciones se transforme en una suma, lo cual tiene beneficios. En primer lugar, las propiedades asintóticas de las sumas son más simples de analizar, tal como la Ley de Grandes Números y el Teorema Central del Límite[14, 15]. En segundo lugar, las sumas son más estables numéricamente que las multiplicaciones, lo cual es de gran importancia computacionalmente.

Tomando el conjunto de datos de trabajo, y con las anomalías agregadas como se vio en la figura 5.1 se procedió a realizar los pre-procesos mencionados en los capítulos anteriores. En primer lugar se separaron los datos en dos conjuntos: uno de entrena-

miento compuesto por los datos anteriores a la introducción de las anomalías; y uno de validación que abarca los datos con las anomalías ya incluidas.

Para el conjunto de entrenamiento se realizaron los siguientes procesos:

1. Se resamplearon los datos cada tres horas.

2. Se realizó *multivariate trimming* para obtener una transformación más robusta a partir de un mejor estimador de la matriz de correlación.

3. Se estandarizaron los datos.

4. Se separaron los datos según la estación del año en Invierno, Primavera, Otoño y Verano.

5. Se hallaron los autovectores y autovalores de la matriz de correlación de los datos y se transformaron los datos a ese espacio.

6. Para cada estación se encontraron los parámetros de las gaussianas que mejor ajusten los datos según el método EM. Para saber el número de distribuciones óptimo, se utilizó el método la puntuación *Silhouette*, como se ve en la figura 5.2.

7. Se calculó la función *likelihood* para cada punto de entrenamiento. Se encontró la distribución de estos valores para cada estación y se calcula el cuantil que comprenda un porcentaje de los datos deseados. Este valor será el umbral para definir un nuevo punto como nominal o anómalo. Esto se presenta en la figura 5.3.

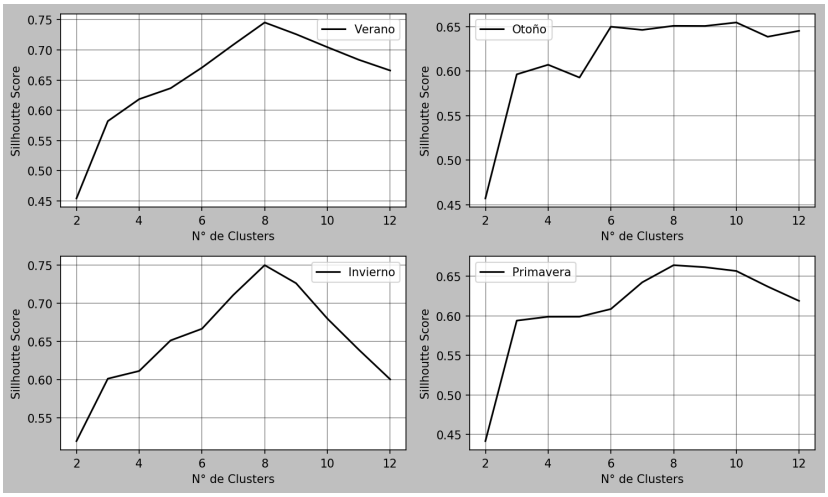


Figura 5.2: Puntuación *Silhouette* para determinar el número de *Clusters* a realizar.

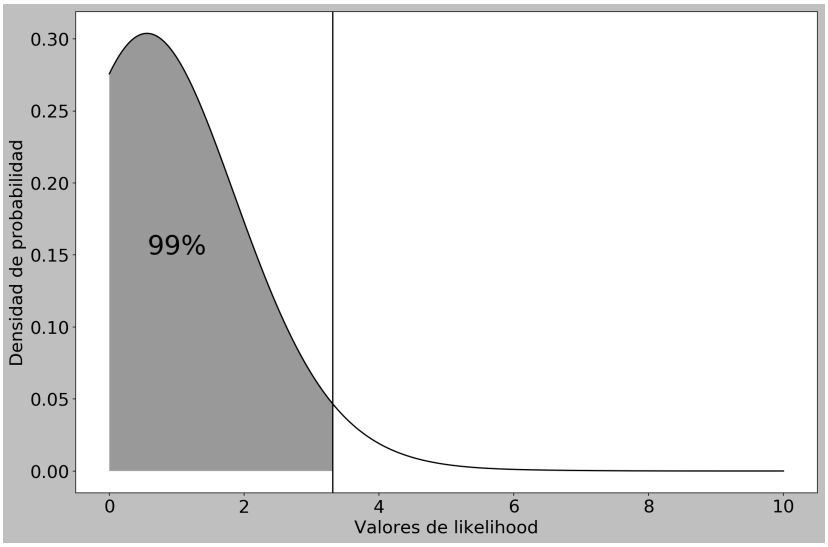


Figura 5.3: Distribución de los valores de *likelihood* para los datos de entrenamiento. Umbral del 99 %.

Para los datos de validación se procedió de forma similar:

1. Se resamplearon los datos cada tres horas.

2. Se estandarizaron los datos.

3. Se proyectaron al espacio de Componentes Principales antes encontrados.

4. Se calculó la función *likelihood* para cada punto de validación.

5. Se comparó el valor de *likelihood* contra el umbral antes encontrado y se lo catalogó como nominal o anómalo. Para el conjunto de datos que contienen las anomalías mostradas en la figura 5.1 se obtuvieron los resultados mostrados en la figura 5.4.

Comparando la figura 4.11, la cual pertenece al mismo periodo de tiempo y no tiene anomalías, con la figura 5.4, se puede ver un corrimiento de los datos. Esto provoca que una parte de los datos comience a quedar por fuerza de las zonas esperadas y sean detectados como anomalías.

5.1.2. PCC

Con las mismas anomalías generadas se procedió a validar el algoritmo PCC. Al igual que en el caso anterior, se dividió el conjunto de datos desde el punto en que comienzan las anomalías.

Para el conjunto de entrenamiento se realizaron los siguientes procesos:

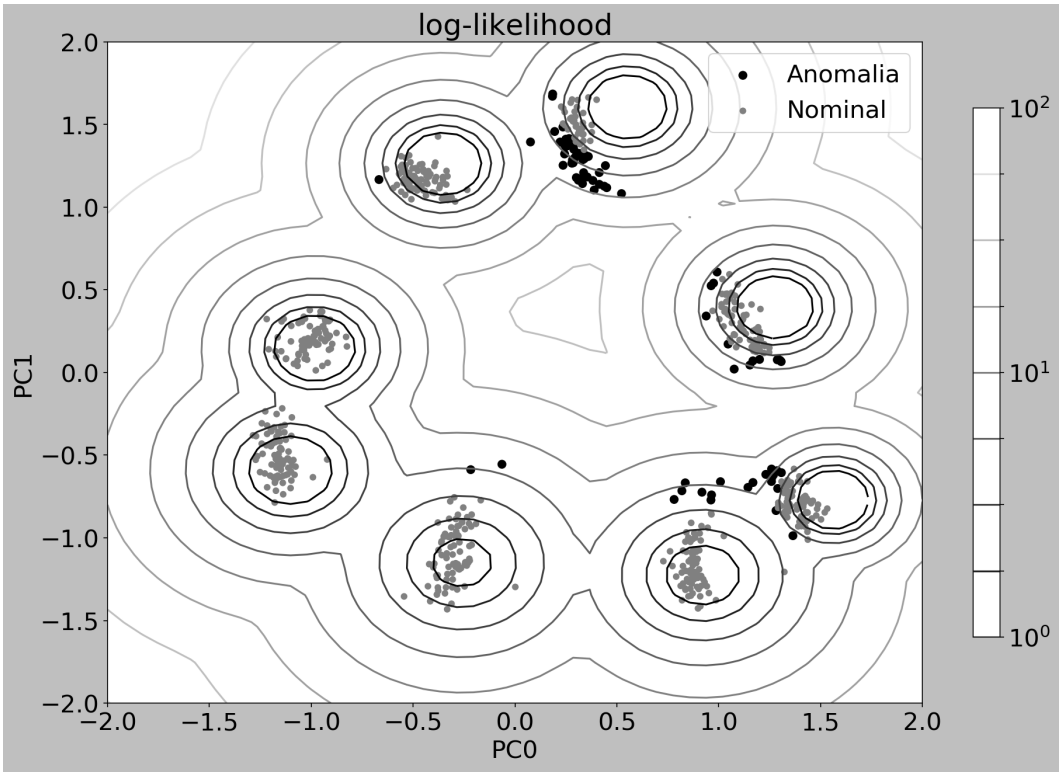


Figura 5.4: Clustering con GMM. Diferenciación entre puntos anómalos artificiales y nominales, para una estación del año específica.

1. Se resamplearon los datos cada una hora.
2. Se realizó *multivariate trimming* para obtener una transformación más robusta a partir de un mejor estimador de la matriz de correlación.
3. Se guardaron la media y el desvío estándar de los datos de entrenamiento y se estandarizaron los datos.
4. Se transformaron al espacio de los Componentes Principales.
5. Se halló C_1 y C_2 como se mostró en la sección 4.1.3 para un umbral del 99 %.

Luego, con los datos modificados se realizaron los siguientes pasos:

1. Se resamplearon los datos cada una hora.
2. Se los estandarizó con la media y desvío estándar de las variables de entrenamiento.
3. Se transformaron al espacio de los Componentes Principales de los datos de entrenamiento, es decir, usando los vectores encontrados anteriormente.
4. Se calculó la suma de los cuadrados de los elementos de los primeros y últimos componentes principales según la ecuación 4.16. Se tomó como primeros Componentes a aquellos que explican menos del 70 % de la varianza total, y los últimos

como aquellos que individualmente expliquen menos del 1 %. Estos valores elegidos se basan en experiencia empírica y en recomendaciones de [6].

5. Se definió cada punto como nominal o anómalo como se explicó en la sección 4.1.3.

Se realizó un histograma donde se agruparon cada 24 puntos (dado el período de una hora, esto representa cada un día) el número de anomalías que se contabilizaron. Durante el crecimiento de las temperaturas, el método detectó el 100 % de los puntos como anómalos. Esto puede verse en la figura 5.5. En contraste, se realizó el mismo proceso para los mismos datos si no se hubiesen agregado las anomalías artificiales, y se encuentra un porcentaje de puntos anómalos de alrededor del 1 %. Esto es acorde al ruido esperado dado del umbral seleccionado para C_1 y C_2. Esto último se presenta en la figura 5.6

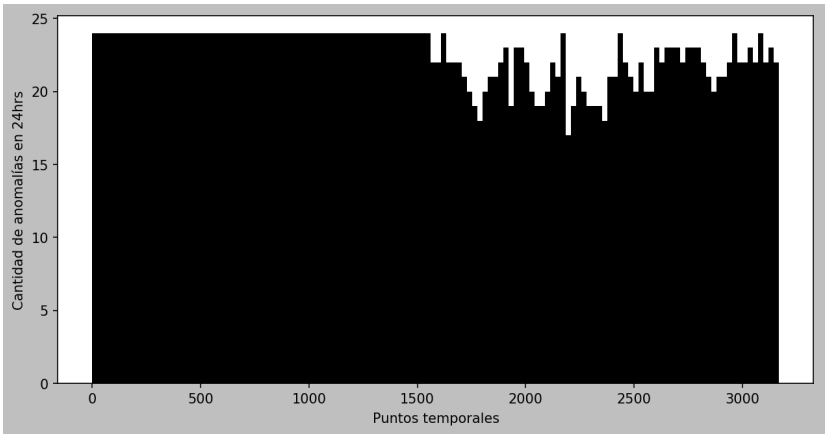


Figura 5.5: Histograma donde se agruparon cada 24 puntos el número de anomalías que se contabilizaron en el conjunto de datos con anomalías artificiales.

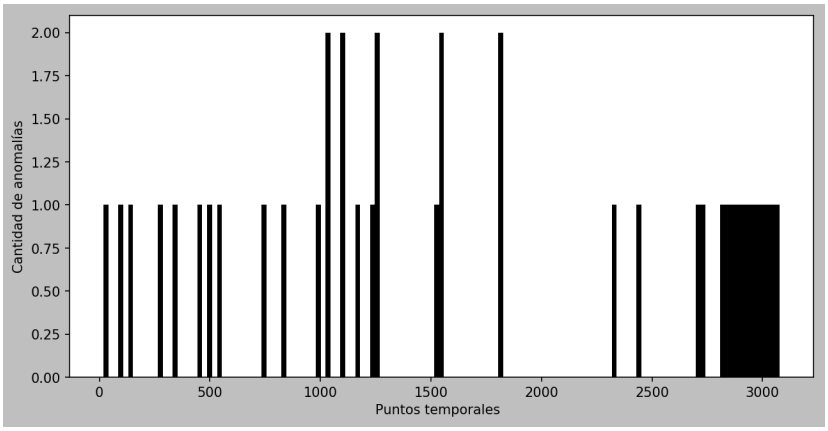


Figura 5.6: Histograma donde se agruparon cada 24 puntos el número de anomalías que se contabilizaron en el conjunto de datos de la figura 5.5 sin anomalías artificiales.

5.2. Anomalía artificial: Puntual

Luego se propuso utilizar anomalías puntuales. Para ello se calcula el desvío estándar en la cercanía al punto que se quiere hacer anómalo. Se multiplica esto por un factor arbitrario y se lo suma al punto en cuestión. El resultado de esto aplicado a variables de temperatura se puede ver en la figura 5.7. Se agregaron siempre a la misma hora, una vez por día durante un mes.

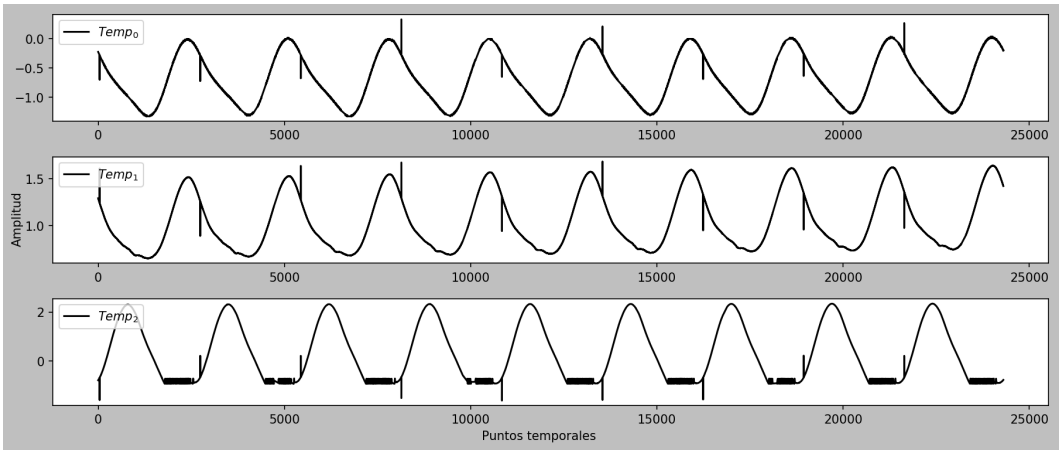


Figura 5.7: Anomalías artificiales puntuales agregadas a variables de temperatura.

5.2.1. Clustering

Siguiendo los pasos ya planteados en la sección 5.1.1 se analizaron los nuevos datos de validación y se obtuvo la figura 5.8. Es difícil saber si las anomalías que se observan son debido al ruido o efectivamente se detectan los puntos anómalos creados anteriormente. El número porcentual de anomalías es de aproximadamente el 5 %, lo cual está por encima del umbral esperado si fuera solo debido al ruido. Este método no es concluyente en este caso debido a la falta de sensibilidad del método frente a anomalías puntuales. *Gaussian Mixture Model* refleja mejor aspectos integrales de los datos como lo son cambios en la estructura de los mismos.

Se cree que el desvío respecto a los centroides de las distribuciones que se aprecia en los datos está dado principalmente por un decaimiento sistemático en la medición de temperatura. Esto es inherente a los sensores utilizados y por lo tanto es característico de este conjunto de datos en particular.

5.2.2. PCC

Análogamente a lo que se planteó en la sección 5.1.2 se obtienen los resultados que se muestran en la figura 5.9. En este caso, el algoritmo es más susceptible a las

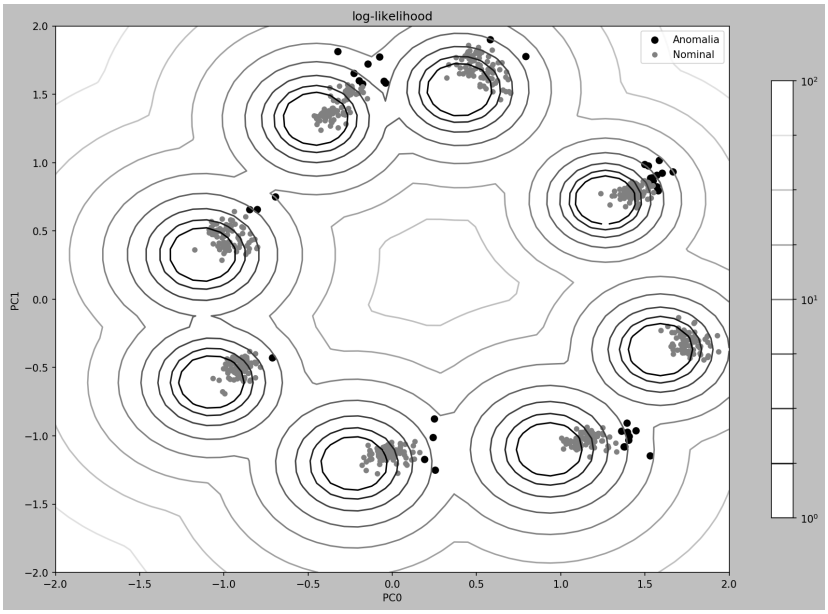


Figura 5.8: Anomalías artificiales puntuales agregadas a variables de temperatura analizadas con GMM.

anomalías puntuales que *Gaussian Mixture Model*. Más precisamente es capaz de detectar anomalías contextuales, aquellas que pueden ser consideradas como anómalas de acuerdo al contexto específico. Los valores agregados artificialmente no sobrepasan los típicos valores que pueden tomar las variables, sino que están fuera de contexto.

Al igual que en el caso de las tendencias impuestas como anomalías artificiales, es notable que el método de *Principal Component Classifier* note las anomalías desde un primer momento y no sea necesaria una acumulación de eventos anómalos.

Por otro lado, la no uniformidad en la detección de anomalías en el período de estudio, se debe a que los valores que se agregaron no fueron constantes. Dado que los valores agregados dependen del desvío estándar de la variable, esto puede oscilar de acuerdo a que punto se tome y a su entorno.

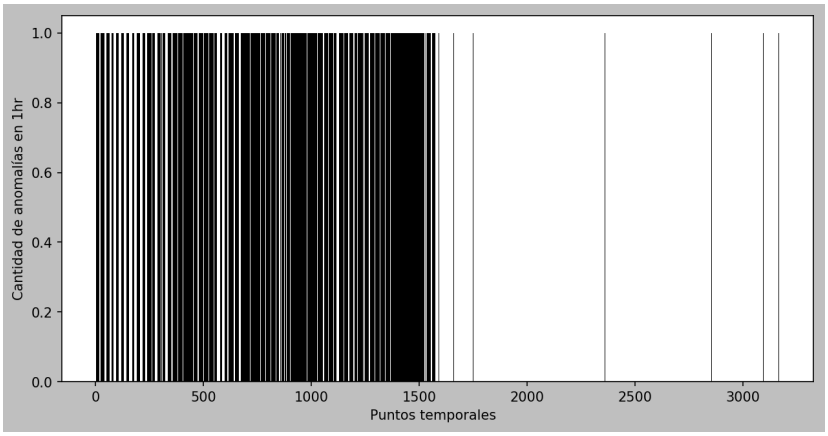


Figura 5.9: Anomalías artificiales puntuales agregas a variables de temperatura analizadas con PCC.

5.3. Anomalías Naturales

Una vez que se probaron los modelos y se obtuvo información de su comportamiento y robustez, se procedió a analizar anomalías naturales de los datos. Para ello se continuó trabajando con el mismo conjunto de datos y se agregó uno nuevo de forma de complementar el estudio. Este último grupo de datos proviene de una plataforma muy similar en estructura a la anterior pero no tanto en las condiciones del entorno, como se verá enseguida en los datos. Analizando la distancia de Mahalanobis de los datos originales se encontró un claro cambio en la estructura de los mismos. Esto se muestra en la figura 5.10. Puede verse que aparece una alteración abrupta hacia el final de los datos, donde aumenta la distancia de Mahalanobis para todos los puntos. Efectivamente, se informó que para el tiempo en que se detecta ese cambio en la estructura, hubo un cambio adrede de configuración en la plataforma de estudio, lo cual valida lo observado en los datos. Este tipo de detección es coherente con el objetivo de la metodología en donde no se buscan anomalías individuales de cada variable, sino que un cambio en la plataforma en su conjunto.

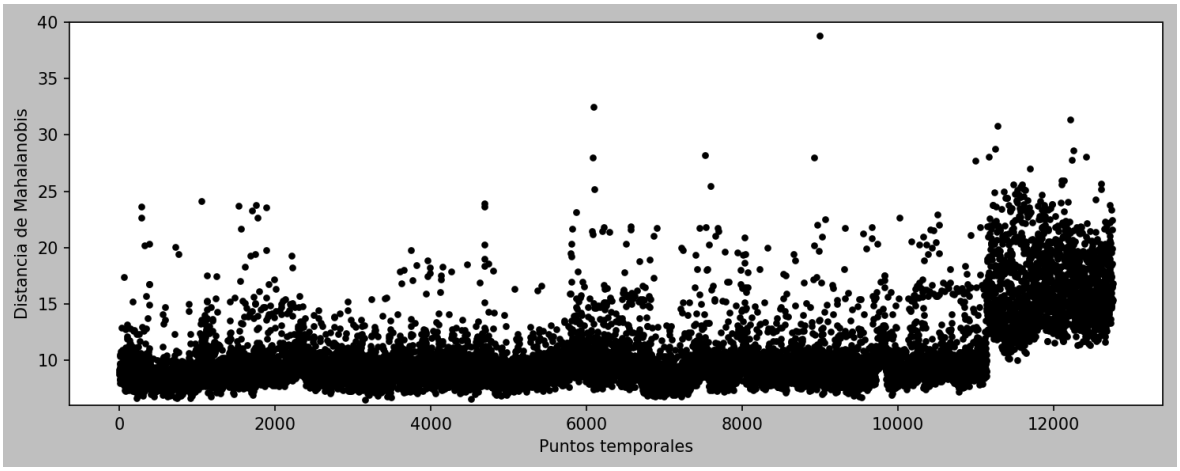


Figura 5.10: Distancia de Mahalanobis de los datos de trabajo en donde se aprecia un claro cambio en la estructura de los mismos.

Para la nueva plataforma se encontró que, aunque es muy similar a la primera, la estructura de los datos es considerablemente distinta a simple vista. Realizando el mismo análisis que para el primer caso, se realizan los pre-procesamientos vistos con anterioridad transformando los datos al espacio de PCA, de acuerdo a la estación que pertenezcan. Esto se puede ver en la figura 5.11. Aunque esto restringe la utilización del método de *Clustering*, dado que no hay una estructura nominal clara de la cual aprender, es esta misma la razón que lleva a entender que existe algún tipo de anomalía en la plataforma. Por la experiencia previa con datos muy similares, estas metodologías exponen un aspecto más caótico de este conjunto de datos, lo cual puede considerarse anómalo en sí mismo.

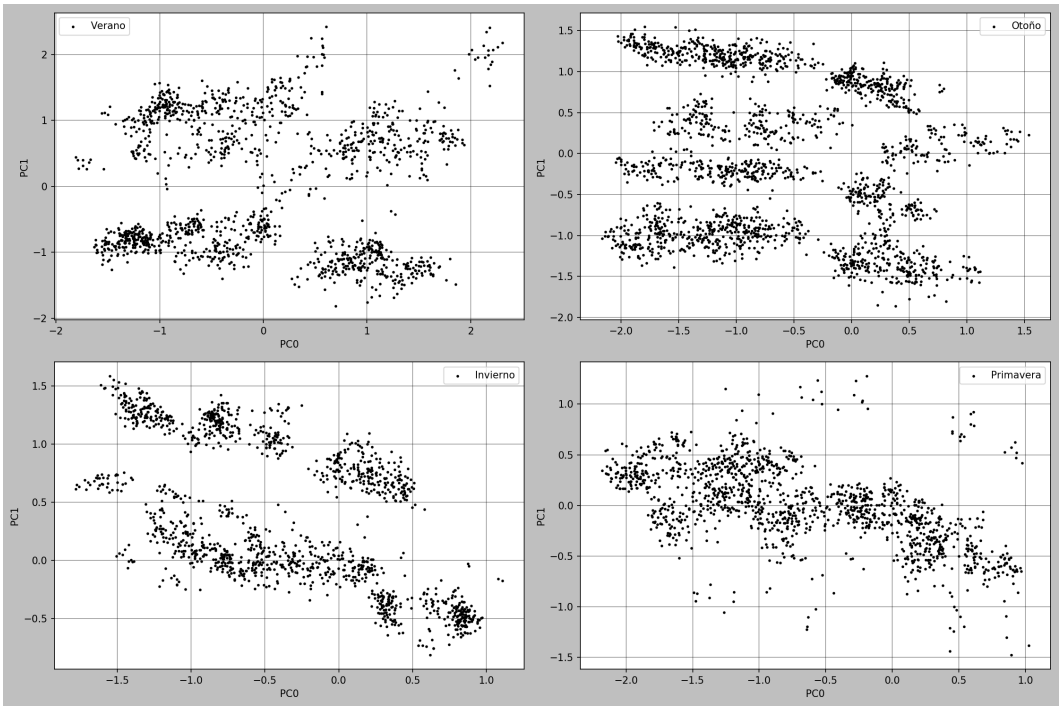


Figura 5.11: Transformación de los nuevos datos a su espacio de PCA. Estos nuevos datos aparecen sin la estructura antes vista aún proviniendo de una plataforma similar.

Por último, mirando la distancia de Mahalanobis de este último conjunto de datos se observan varios grupos temporales que se encuentran significativamente por encima de la media. Se presenta esto en la figura 5.12. En comparación con la figura 4.6, que muestra la distancia de Mahalanobis para el conjunto original, se observan más cantidad de datos que sobresalen del conjunto individualmente, pero fundamentalmente se observan grupos anómalos. Estas observaciones realizadas son acordes a información contrastada de la plataforma.

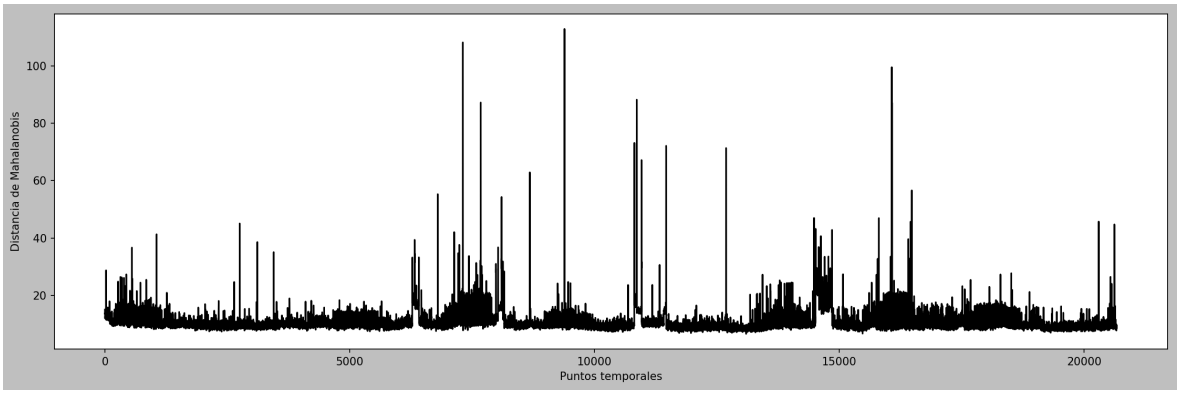


Figura 5.12: Distancia de Mahalanobis para los nuevos datos. Se ven grupos temporales que se encuentran notoriamente por encima de la media.

Capítulo 6

Conclusiones

> *"Dime y lo olvido, enséñame y lo recuerdo, involúcrame y lo aprendo."*
>
> — Benjamin Franklin, 1706-1790

El objetivo de este proyecto integrador fue identificar e implementar algoritmos de aprendizaje para la detección de anomalías. Se plantearon herramientas integrales, *Clustering* y *PCC*, que tienen el propósito de analizar varias variables al mismo tiempo y tienen en cuenta la estructura de los datos en su conjunto. Además, se presentó la herramienta de *Forecasting* para un análisis de las variables individuales, realizando una predicción de su comportamiento. En todos los casos son herramientas que se pensaron para ayudar a complementar el análisis del profesional en su trabajo, y no ser utilizadas independientemente. Son herramientas versátiles y suficientemente simples en su concepto como para ser aplicadas a distintos conjuntos de datos. Inclusive la estructura y formato de presentación permite que se pueda aplicar ágilmente a nuevos conjuntos de datos.

El conjunto de datos presentaba diferentes tipos de variables. Para mantener el análisis y el enfoque lo más general posible se evitó un tratamiento particular sobre variables específicas, todas ellas, dentro de límites razonables, fueron tratadas en una base de igualdad. Se trataron los faltantes de datos con interpolaciones lineales, y se trabajaron todos los modelos a partir de ello.

Antes de comenzar con los modelos se presentaron paradigmas del aprendizaje, los cuales moldearon el enfoque a lo largo del proyecto. Son ideas, aunque simples, críticas a la hora de desarrollar los modelos.

Un caso donde fue de relevancia el principio de Occam, fue para el ajuste de los parámetros de las gaussianas en *Gaussian Mixture Model*. La teoría dice que al ser variables descorrelacionadas en el espacio de PCA, los términos de las gaussianas deben ser independientes entre sí en cada dimensión. Sin embargo, si se permiten más grados de libertad, de manera que no haya simetría en ninguno de los ejes, se obtiene un mejor

ajuste, o un mejor *in-sample error*. Esto puede verse en la figura 6.1, en comparación a la obtenida previamente en la figura 4.11. Es decir, la probabilidad de cada dato de haber sido generado por las gaussianas es más alto si se le dan más grados de libertad al modelo. Esto va en contra del principio de simpleza de los modelos y muy probablemente se incurra en *Overfitting*. El modelo tiene que ser lo más simple posible, de esta manera se asegura que generalice mejor a datos nuevos.

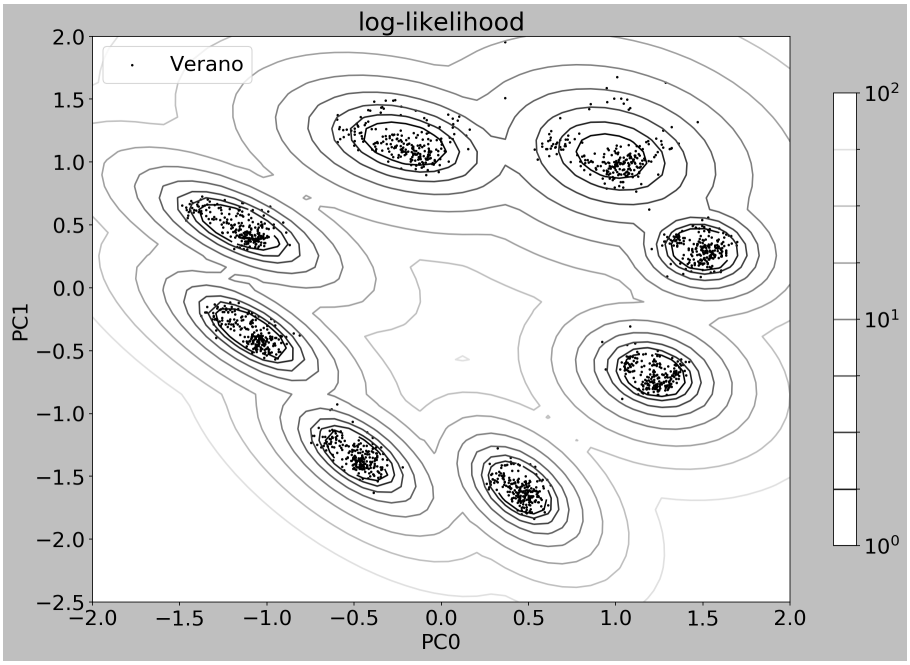


Figura 6.1: *Gaussian Mixture Model* con los términos de varianza y covarianza libres de ser ajustados. Modelo más complejo y con mejor error *in-sample*. Esto va en contra del principio de simpleza de los modelos y muy probablemente se incurra en *Overfitting*.

Realizar un mejor estimador de la matriz de correlación surgió en base a que se hallaba mucho ruido entre las matrices de correlación de los mismos datos pero para distintos períodos de tiempo. Dado que la matriz de correlación poblacional es única para un dado conjunto de datos, que esta variara según el período de tiempo marcaba que la matriz muestral estaba estimando incorrectamente. Tras realizar la metodología explicada en la sección 4.1.2 se halló un mejor estimador que representaba adecuadamente los datos.

Para los resultados presentados a partir de anomalías artificiales se encontró que en *Clustering* una tendencia impuesta de temperatura desplaza el conjunto de datos completo dentro del espacio de PCA. Esto tiene como consecuencia que si el desplazamiento no es lo suficientemente grande, los puntos se sigan encontrando dentro de la holgura que les da la campana gaussiana. Esto se vio en la figura 5.4, donde en particular hay un solo *Cluster* donde el conjunto de puntos comenzó a estar significativamente por fuera del umbral impuesto. Sin embargo, para los demás *Clusters* esto no fue así, y a pesar de que se desplazaron, no fue suficiente para traspasar el límite de detección

anómala.

Para el caso de *Principal Component Classifier* se aprecia en la figura 5.5 que marca la existencia de anomalías en el 100 % de los datos durante la tendencia de aumento de temperatura. Mientras que cuando deja de estar la tendencia, el número de anomalías, aunque alto, ya no es en todos los puntos. Esto marca que gran parte de la detección no se genera por la existencia de valores extraño a la media, sino que por un cambio de estructura en los datos, como lo es una tendencia impuesta en varias variables al mismo tiempo.

Comparando estas dos herramientas, *Principal Component Classifier* y *Clustering*, es importante notar que para este último se están descartando gran cantidad de dimensiones y sólo se conservan las primeras dos componentes de PCA. Esto trae consigo una falta de sensibilidad ante pequeñas perturbaciones o anomalías puntuales como se observó en la sección 5.2.1. Mientras que el método de *Principal Component Classifier* tiene en cuenta la mayoría de las dimensiones, siendo más sensible a más tipos de anomalías, como lo son los eventos puntuales o tendencias en las variables. Esto no hace que uno sea mejor que el otro, sino que el espectro de aplicación puede ser diferente y posiblemente complementario.

Además, se notó en los datos una tendencia leve pero capaz de desplazar los datos lo suficiente en el espacio de PCA entre temporadas como para ser detectados como anomalías. Tomando esta tendencia natural como nominal y esperada, puede tratar de contemplarse en los modelos. En el caso de GMM, la dispersión de los datos (la varianza) se mantiene constante, pero los centroides de las distribuciones estan en continuo movimiento. Con los suficientes datos, puede tratar de predecirse la tendencia y realizar un ajuste año a año de los parámetros de las gaussianas.

Adicionalmente, se observan dependencias no lineales entre los datos que pueden ser explotadas con metodologías más complejas como lo pueden ser algunos modelos de Redes Neuronales.

Por último se enfatiza el modo de operación de las herramientas desarrolladas en este trabajo, en donde se proponen con un fin de facilitar el trabajo del encargado del análisis de las telemetrías, y en ningún caso pretenden funcionar de forma autónoma. Es necesario ajustar los hiper-parámetros en la etapa de entrenamiento para cada conjunto de datos que se analice, y por lo tanto es necesaria cierta experiencia del entrenador.

Bibliografía

[1] B. Shetty, "Curse of dimensionality," Jan 2019. vii, 21

[2] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data: a short course*. AMLbook.com, 2012. 1, 13

[3] C. M. BISHOP, *PATTERN RECOGNITION AND MACHINE LEARNING*. SPRINGER-VERLAG NEW YORK, 2016. 1

[4] R. H. Jones, "Entrepreneurs, beware survivorship bias," Mar 2019. 14

[5] B. Clarke, F. Ernest, and H. H. Zhang, *Principles and theory for data mining and machine learning*. Springer, 2011. 21

[6] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," Jan 2003. 23, 31, 48

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. 23, 35, 36, 37

[8] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Versión: 1.2.1]. 28

[9] J. D. Jobson, *Applied Multivariate Data Analysis Categorical and Multivariate Methods*, vol. 2. Springer Verlag, 2013. 30

[10] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010. 39

[11] R. J. Hyndman and G. Athanasopoulos, "Forecasting: Principles and practice, 2nd edition," Apr 2018. [Accedido: Abril 2019]. 39

[12] R. Cleveland and W. Cleveland, "Stl: A seasonal-trend decomposition procedure based on loess," *Journal of Official ...*, vol. 6, 01 1990. 40

[13] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. 40

[14] L. Breiman, *Probability and stochastic processes: with a view toward applications*. Scientific Press, 1986. 44

[15] P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to probability theory*. Houghton Mifflin, 1996. 44

Agradecimientos

A mis padres, por el apoyo incondicional a lo largo de la vida.

A Graciela, por su predisposición extraordinaria en todo momento.

A José, por ser un excelente mentor y guiarme a lo largo de este proyecto.

A Félix, por su entusiasmo y motivación sobresalientes en el desarrollo del trabajo.

A Sara, por acompañarme a lo largo de este viaje.

Gracias.